



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

NICOLAS DE ALMEIDA MARTINS

**JOGOS DE PERSEGUIÇÃO EM GRAFOS E COLORAÇÃO
LOCALMENTE IDENTIFICÁVEL**

FORTALEZA

2018

NICOLAS DE ALMEIDA MARTINS

JOGOS DE PERSEGUIÇÃO EM GRAFOS E COLORAÇÃO LOCALMENTE
IDENTIFICÁVEL

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Departamento de Computação da Universidade Federal do Ceará, como parte dos requisitos necessários para a obtenção do título de Doutor em Ciência da Computação. Área de concentração: Algoritmos e Otimização.

Orientador: Prof. Dr. Rudini Menezes Sampaio

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- M344j Martins, Nicolas de Almeida.
Jogos de Perseguição em Grafos e Coloração Localmente Identificável / Nicolas de Almeida Martins. –
2018.
93 f. : il. color.
- Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em
Ciência da Computação, Fortaleza, 2018.
Orientação: Prof. Dr. Rudini Menezes Sampaio.
1. Teoria dos Grafos. 2. Coloração de Grafos. 3. Jogos de Perseguição. I. Título.

CDD 005

NICOLAS DE ALMEIDA MARTINS

JOGOS DE PERSEGUIÇÃO EM GRAFOS E COLORAÇÃO
LOCALMENTE IDENTIFICÁVEL

Tese apresentada ao Programa de Pós-graduação em Ciência da Computação do Departamento de Computação da Universidade Federal do Ceará, como parte dos requisitos necessários para a obtenção do título de Doutor em Ciência da Computação. Área de concentração: Algoritmos e Otimização.

Aprovada em: 22 / 01 / 2018.

BANCA EXAMINADORA

Prof. Dr. Rudini Menezes Sampaio (Orientador)
Universidade Federal do Ceará (UFC)

Prof^a Dra. Ana Karolinnna Maia de Oliveira
Universidade Federal do Ceará (UFC)

Prof^a Dra. Ana Shirley Ferreira da Silva
Universidade Federal do Ceará (UFC)

Prof. Dr. Ronan Pardo Soares
Universidade Federal do Ceará (UFC)

Prof^a Dra. Sulamita Klein
Universidade Federal do Rio de Janeiro (UFRJ)

Dedico este trabalho a todas as pessoas que
contribuíram direta ou indiretamente com a
sua realização.

AGRADECIMENTOS

Várias pessoas contribuíram para que este trabalho fosse realizado. Sou profundamente grato a todas, e dedico agradecimento especial a algumas delas:

À minha esposa, Carliane, e minha filha, Cecília, tudo que faço na minha vida só faz sentido pois vocês duas existem. Os seus sorrisos foram o combustível que me deu força para concluir este trabalho.

Aos meus pais, Evandro e Silvia, por terem me apoiado por toda a minha vida. Vocês são os melhores pais que eu poderia desejar.

A meu orientador e amigo, Rudini Sampaio, que me atura desde a graduação e que, além de ser um excelente orientador e um cientista brilhante, ainda organiza as partidas de futebol de salão para que os alunos possam aliviar o estresse!

A todos os colegas do Inria Sophia Antipolis, especialmente à Nicolas Nisse e Patricia Lachaume, que me acolheram e me fizeram sentir em casa.

Aos meus professores e amigos, Ronan e Karol, que com seus conselhos ajudaram imensamente a melhorar a qualidade deste texto.

Gostaria ainda de dedicar um agradecimento especial a um amigo que, infelizmente, perdi recentemente. Fenrir, você estará sempre no meu coração! Você é, e sempre será, meu meninzinho bom!

Finalmente, ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - pelo financiamento da bolsa de estudos para a realização da presente dissertação.



“If you think it’s simple, then you have misunderstood the problem.”

Bjarne Stroustrup

RESUMO

Nesta tese estudamos os problemas de coloração localmente identificável (lid-coloração) e diversos parâmetros relacionados a jogos de perseguição em grafos. Para colorações localmente identificáveis, mostramos que o número lid-cromático e o número lid-cromático forte são ambos $O(n^{1-\varepsilon})$ -inaproximáveis em tempo polinomial, a menos que $P = NP$. Também mostramos algoritmos lineares para $(q, q - 4)$ -grafos e para grafos com largura em árvore limitada para ambos os parâmetros. Com relação a jogos de perseguição, estudamos dois jogos diferentes: O Jogo de Polícia e Ladrão e o Jogo do Espião. Para o Jogo de Polícia e Ladrão mostramos valores exatos para o *cop-number* e o *k-capture time* de grafos P_4 -tidy e $(q, q - 4)$ -grafos. Além disso mostramos que a famosa conjectura de Meyniel é válida para grafos P_4 -tidy conexos e $(q, q - 4)$ -grafos conexos com pelo menos q vértices. Com relação ao Jogo do Espião mostramos que este problema é NP-difícil para qualquer velocidade s e qualquer distância d e, além disso, $(1 - o(1)) \ln n$ -inaproximável em tempo polinomial, a menos que $P = NP$. Ademais, mostramos limites para o número guardas necessários para vencer o jogo quando o mesmo transcorre em ciclos e caminhos. Provamos ainda que a versão direcionada do jogo é PSPACE-difícil para DAG's.

Palavras-chave: lid-Coloração. Jogo de Polícia e Ladrão. Jogo do Espião. Decomposição Primeval. Inaproximabilidade. Cop-Number. Capture Time.

ABSTRACT

On this thesis we study the locally identifying coloring of graphs (lid-coloring) and many parameters related to pursuit games on graphs. For locally identifying colorings, we show that the lid-chromatic number and the strong lid-chromatic number are both $O(n^{1-\varepsilon})$ -inapproximable in polynomial time, unless $P = NP$. We also show linear algorithms for $(q, q - 4)$ -graphs and for graphs with bounded treewidth for both parameters. Regarding pursuit games, we study two distinct games: The Cops and Robber and the Spy-Game. For the Cops and Robber game we show exact values for the *cop-number* and the *k-capture time* of P_4 -tidy graphs and $(q, q - 4)$ -graphs. Furthermore we show that the famous Mayniel conjecture is true for P_4 -tidy graphs and $(q, q - 4)$ -graphs with at least q vertices. Regarding the Spy-Game we show that it is NP-hard for any speed s and any distance d and, besides that, it is $(1 - o(1)) \ln n$ -inapproximable in polynomial time, unless $P = NP$. Furthermore, we show bounds to the number of guards needed to win the game when it is played on paths and cycles. We also prove that the directed version of the game is PSPACE-hard for DAG's.

Keywords: lid-Coloring. Cops and Robber. Spy-Game. Primeval Decomposition. Inapproximability. Cop-Number. Capture Time.

LISTA DE FIGURAS

Figura 1 – Exemplo de representação estrutural do mapa do Nordeste Brasileiro . . .	12
Figura 2 – Exemplo de representação de um grafo	19
Figura 3 – Mapa de uma cidade e sua representação em um grafo	21
Figura 4 – Exemplo de representação de um digrafo	22
Figura 5 – Aranha Magra	24
Figura 6 – Aranha Gorda	24
Figura 7 – Exemplo de um grafo e sua respectiva árvore de decomposição	26
Figura 8 – Grafos P_4 -tidy e suas subclasses	27
Figura 9 – Exemplo de Decomposição em Árvore	28
Figura 10 – Exemplos de lid-colorações ótimas	32
Figura 11 – Coloração própria de K_4^*	33
Figura 12 – Lid-coloração ótima de K_4^*	33
Figura 13 – Exemplo de redução para o grafo de Grötzsch. Apresentamos uma coloração própria de G e uma lid-coloração de G'	37
Figura 14 – Exemplo de G e G'	38
Figura 15 – Exemplo de Coloração Própria de G e lid-Coloração de G'	39
Figura 16 – Exemplos de grafos <i>cop-win</i>	52
Figura 17 – Grafo com vértice universal	52
Figura 18 – Árvore	52
Figura 19 – Grafo Estrela	52
Figura 20 – Exemplo de grafo <i>dismantable</i> com a ordenação dos vértices	53
Figura 21 – Exemplo de p-componente separável (módulos em destaque) e sua respectiva p-componente característica	61
Figura 22 – p-componente separável	61
Figura 23 – p-componente característica	61
Figura 24 – Redução da instância (\mathcal{S}, c) do problema de cobertura de conjuntos, onde $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ e $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Casos para velocidade $s = 5$ e distância $d = 2, 3$. Ilustração da prova do Lema 5.1	68
Figura 25 – Redução da instância (\mathcal{S}, c) do problema de cobertura de conjuntos, onde, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ e $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Casos para velocidade $s = 5$ e distância $d \in \{0, 1\}$. Ilustrações da prova do Lemma 5.2	69

Figura 26 – Redução de uma instância (\mathcal{S}, c) do problema de cobertura de conjuntos, onde $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ e $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Casos para velocidade $s = 5$ e distância $d \in \{4, 5\}$. Ilustração das provas do Lema 5.4 (esquerda) e Lema 5.5 (direita)	71
Figura 27 $\neg D_i(\exists)$	75
Figura 28 $\neg D_i(\forall)$	75
Figura 29 $\neg D_\phi$, onde $\phi = \exists x_1 \forall x_2 (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2)$	75
Figura 30 – Esquema de posições para o caso em que $k = 8$, $s = 2$	81

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Coloração Localmente Identificável	13
1.2	Jogo de Polícia e Ladrão	15
1.3	Jogo do Espião	17
2	CONCEITOS BÁSICOS	19
2.1	Grafos	19
2.2	Grafos Direcionados	21
2.3	Grafos com poucos P4's	22
2.4	Grafos com Largura em Árvore Limitada	28
2.5	Tratabilidade com Parâmetro Fixo	29
2.6	Algoritmos Aproximativos e Inaproximabilidade	30
3	COLORAÇÃO LOCALMENTE IDENTIFICÁVEL	32
3.1	Introdução	32
3.2	Resultados Preliminares	34
3.3	Inaproximabilidade dos parâmetros de lid-coloração	36
3.4	Lid-Colorações de Grafos com Poucos P4's	40
3.5	Lid-Coloração em grafos com largura em árvore limitada	47
4	JOGO DE POLÍCIA E LADRÃO EM GRAFOS	51
4.1	Introdução	51
4.2	Parâmetros de captura em grafos com poucos P4's	55
5	JOGO DO ESPIÃO	63
5.1	Introdução	63
5.2	NP-Compleitude e Inaproximabilidade	67
5.3	Complexidade para o caso direcionado	73
5.4	Limites para caminhos e ciclos	77
5.5	Jogo do espião em grafos P4-tidy	82
6	CONCLUSÃO	88
	REFERÊNCIAS	89

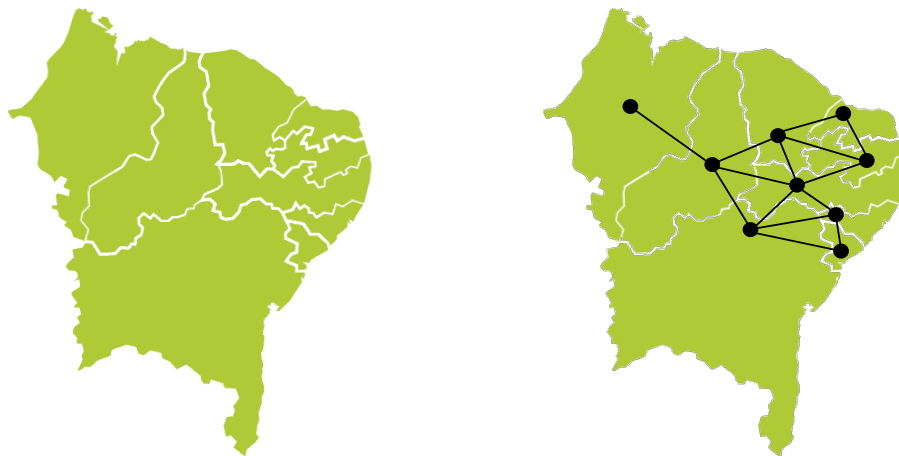
1 INTRODUÇÃO

Nesta tese, estudamos os problemas de coloração localmente identificável (lid-coloração) e alguns parâmetros relacionados a dois jogos de perseguição em grafos.

A história dos problemas de coloração em grafos tem sua origem na famosa *Conjectura das Quatro Cores* proposta por Morbiüs em 1840. Nela Morbiüs afirmava que todo mapa pode ser colorido com no máximo quatro cores sem que regiões vizinhas possuam a mesma cor. Doze anos depois, Guthrie formulou a mesma conjectura de maneira independente.

Em 1879, Kempe observou que poderia, a partir de qualquer mapa, obter sua representação estrutural ao transformar cada região em um ponto e ligar os pontos que representam regiões vizinhas por curvas. Assim o problema de colorir as regiões de um mapa é equivalente a colorir os pontos de sua representação estrutural de modo que dois pontos ligados por uma curva sempre tenham cores distintas. A representação com pontos e curvas ficaria, mais tarde, conhecida como grafo, onde os pontos são os vértices e as curvas são as arestas. Neste caso grafos planares, pois podem ser representados no plano sem cruzamento entre as arestas. A atribuição de cores que distingue dois vértices adjacentes ficou conhecida como *Coloração Própria* de vértices.

Figura 1 – Exemplo de representação estrutural do mapa do Nordeste Brasileiro



Fonte: Elaborado pelo autor.

A Conjectura das Quatro cores permaneceu em aberto até 1976 quando Kenneth Appel e Wolfgang Haken provaram a sua validade utilizando o método da descarga em, pela primeira vez na história, uma prova auxiliada por computador. Desde então muitas variantes do problema de coloração em um grafo surgiram. Podemos citar dentre essas os problemas de *Coloração de Arestas*, *Coloração completa*, *Coloração Gulosa*, *Coloração com Listas*, etc.

Jogos de perseguição também tem sua origem séculos atrás, em 1732, com Pierre Bouguer estudando o problema de um navio pirata perseguindo um navio mercante

para saqueá-lo.

Em grafos, tais jogos consistem de, normalmente, dois jogadores que, seguindo regras específicas de cada jogo, movimentam, adicionam ou removem peças dos vértices ou arestas de um grafo. Jogos desse tipo tem inúmeras aplicações como, por exemplo, simular a busca por um terrorista escondido em um sistema de túneis e cavernas ou mesmo um vírus se propagando por uma rede de computadores.

Normalmente, ao estudarmos jogos de perseguição visamos obter o menor número de recursos necessários para que o “jogador perseguidor” sempre tenha êxito em sua busca, independente da estratégia adotada pelo “jogador em fuga”.

Ao abordarmos estes temas, iremos nos concentrar, especialmente, no estudo de suas complexidades. Todos os problemas abordados são NP-difíceis e, portanto, é pouco provável que existam algoritmos polinomiais para solucioná-los para o caso geral. Assim, iremos estudar a complexidade destes problemas quando restringimos o conjunto de entradas a grafos com certas propriedades. Um exemplo deste tipo de abordagem é o problema de coloração própria de vértices que, embora seja NP-completo para grafos quaisquer, é polinomial quando os grafos fornecidos como entrada são sempre grafos sem ciclos. Vale ressaltar que restringir as entradas de um problema nem sempre resulta em uma diminuição de sua complexidade: o problema de coloração própria de vértices permanece NP-completo mesmo quando as entradas fornecidas são restritas a grafos planares (GAREY, JOHNSON, and STOCKMEYER, 1976).

Outra forma de estudar um problema de otimização difícil é mostrar se, dada uma entrada arbitrária, podemos ou não obter uma aproximação para o valor de uma solução ótima em tempo polinomial. Um exemplo de aplicação desta abordagem é o problema de *Cobertura por Conjuntos*. Neste, dada uma entrada arbitrária com n elementos e m conjuntos, não podemos encontrar, em tempo polinomial, uma solução viável cujo valor seja no máximo $(1 - o(1)) \ln n$ vezes pior que o valor de uma solução ótima, a menos que $P = NP$ (DINUR and STEURER, 2013).

1.1 Coloração Localmente Identificável

Problemas de coloração estão entre os mais estudados em teoria dos grafos. O interesse neste tipo de problema, iniciado com a Conjectura das Quatro Cores, vem não só do ponto de vista prático como também do ponto de vista teórico.

Do ponto de vista prático, diversos problemas de coloração podem ser utilizados para modelar problemas reais em computação. O problema de coloração própria, por exemplo, pode ser utilizado para solucionar o problema da alocação de registradores em um programa (CHAITIN *et al.*, 1981). A $L(2, 1)$ -coloração é uma coloração cujos vértices a distância 2 devem possuir cores distintas e vértices vizinhos devem possuir cores com pelo menos 2 unidades de diferença. Ela pode ser utilizada, por exemplo, para alocar

frequências em transmissores de modo a evitar interferências (GRIGGS and YEH, 1992).

Do ponto de vista teórico, os problemas de coloração são bastante estudados devido à sua dificuldade. Grande parte destes problemas são NP-completos (GAREY, JOHNSON, and STOCKMEYER, 1976; GRIGGS and YEH, 1992; BODLAENDER, 1989; IRVING and MANLOVE, 1999), o que os tornam importantes para examinar a dicotomia entre as classes de problemas P e NP.

Diversos problemas visam a identificar os vértices de um grafo ou ao menos diferenciar determinados vértices. Podemos citar como exemplo disto os chamados códigos de identificação (KARPOVSKY, CHAKRABARTY, and LEVITIN, 1998) que, utilizando um conjunto dominante do próprio grafo, identificam cada vértice pelos vértices que o dominam. Outro exemplo é o problema de coloração localizadora (locating coloring) (CHARTRAND *et al.*, 2002), que é uma coloração própria dos vértices na qual cada vértice possui um código identificador que depende da sua distância para cada uma das classes de cor.

Recentemente, Esperet *et al.* introduziram mais uma variante do problema de coloração com o objetivo de diferenciar vértices, a chamada lid-coloração (*locally identifying coloring*) (ESPERET *et al.*, 2012). Neste problema, os vértices são identificados pelas cores atribuídas aos vértices de sua vizinhança fechada, de modo que vértices adjacentes devem ser identificados por conjuntos de cores distintos, a menos que as vizinhanças fechadas sejam idênticas.

Obviamente, se atribuirmos uma cor distinta para cada vértice do grafo obtemos uma lid-coloração válida. Sendo assim, o objetivo do problema é utilizar o menor número de cores possível.

Esperet *et al.* mostraram que qualquer grafo bipartido pode ser lid-colorido com no máximo 4 cores (ESPERET *et al.*, 2012). No entanto, no mesmo artigo, eles provaram que o problema de determinar o número mínimo de cores em uma lid-coloração é NP-completo mesmo para grafos bipartidos com grau máximo 3 e cintura fixa, utilizando uma redução do problema de 2-coloração de hipergrafos. Mostraram ainda que o problema é trivial para árvores. Esperet *et al.* ainda conseguiram limitar o número máximo de cores necessárias para colorir k -árvores, grafos de intervalo, grafos split, cografos e grafos periplanares em função do tamanho da maior clique dos grafos (ESPERET *et al.*, 2012).

Para grafos com grau máximo Δ , Esperet *et al.* mostraram que $O(\Delta^3)$ cores são suficientes (ESPERET *et al.*, 2012) e deixaram como questão em aberto decidir se esse limite poderia ser melhorado para $O(\Delta^2)$. Posteriormente, Foucaud *et al.* responderam de maneira afirmativa a esta questão e estabeleceram que $2\Delta^2 - 3\Delta + 3$ cores são suficientes para lid-colorir qualquer grafo com grau máximo Δ (FOUCAUD *et al.*, 2012).

Em 2013, Gonçalves *et al.* mostraram que todo grafo planar pode ser lid-colorido com no máximo 1280 cores (GONÇALVES, PARREAU, and PINLOU, 2013), respondendo outro questionamento deixado em (ESPERET *et al.*, 2012).

No Capítulo 3, estudamos o problema da coloração localmente identificável focando principalmente no aspecto de sua complexidade. Mostramos que o problema é $O(n^{1-\varepsilon})$ -inaproximável em tempo polinomial, a menos que $P = NP$, utilizando uma redução para o problema de coloração própria. Mostramos ainda algoritmos polinomiais para $(q, q - 4)$ -grafos e grafos com largura em árvore limitada, utilizando a decomposição primeval e a decomposição em árvore, respectivamente. A maior parte de nossos resultados foi publicada em (MARTINS and SAMPAIO, 2018).

1.2 Jogo de Polícia e Ladrão

Um dos jogos de perseguição em grafos mais estudados é o jogo de polícia e ladrão introduzido em 1978 por Quilliot em sua tese de doutorado (QUILLIOT, 1978) e posteriormente, de maneira independente, por Nowakowski e Winkler (NOWAKOWSKI and WINKLER, 1983). O jogo transcorre com informação perfeita, isto é, ambos os jogadores sabem das posições ocupadas pelo seu oponente. O jogo é iniciado com o jogador que controla os policiais posicionando suas “peças” nos vértices do grafo, seguido pelo outro jogador, que posiciona o ladrão. A partir de então, os jogadores se revezam, começando pelos policiais, movendo suas peças entre vértices adjacentes do grafo. O jogo termina com vitória dos policiais se a qualquer momento um policial ocupa o mesmo vértice do ladrão. O ladrão vence se consegue evitar sua captura indefinidamente. Ambos os trabalhos (QUILLIOT, 1978; NOWAKOWSKI and WINKLER, 1983) consideraram que o jogador perseguidor possuía apenas uma “peça” (policial) no grafo visando a capturar a “peça” (ladrão) do seu oponente.

Em 1984, Aigner e Fromme introduziram a versão do jogo com vários policiais (AIGNER and FROMME, 1984). Certamente, em um grafo com n vértices, tem-se que n policiais são suficientes para capturar o ladrão, bastando que cada policial ocupe um vértice distinto para assegurar a captura. Então o objetivo do problema passou a ser identificar o menor número possível de policiais que, jogando de forma ótima em um grafo G , sempre tem êxito na captura, independentemente da estratégia utilizada pelo jogador que controla o ladrão. A este parâmetro damos o nome de *cop-number* do grafo, denotado por $c(G)$.

Além de suas aplicações práticas, como simular um vírus em uma rede sendo procurado por programas antivírus, o jogo de polícia e ladrão tem grande apelo teórico. O estudo da complexidade do problema se iniciou com Goldstein e Reingold em 1995 (GOLDSTEIN and REINGOLD, 1995), onde eles mostraram que se o número de policiais não é fixo e se o grafo for direcionado ou as posições iniciais dos jogadores forem fixas o problema é EXPTIME-completo. Goldstein e Reingold ainda conjecturaram que o jogo seria EXPTIME-completo, para um número arbitrário de policiais, mesmo sem as demais restrições. Em 2015, Kinnersley respondeu a conjectura deixada por Goldstein e Reingold

ao provar que o problema é EXPTIME-completo (KINNERSLEY, 2015).

Hahn e MacGillivray mostraram que se o número k de policiais é fixo, então podemos determinar se um grafo G é k -cop-win em tempo polinomial (HAHN and MACGILLIVRAY, 2006).

Em 2010, Fomin et al. mostraram que calcular o número mínimo de policiais necessários para capturar um ladrão é NP-difícil mesmo quando o ladrão possui velocidade $s \geq 1$ (FOMIN *et al.*, 2010). Ademais, eles mostraram que a versão parametrizada do problema, com parâmetro k (o número de policiais), é $W[2]$ -difícil.

Posteriormente, Mamino mostrou que, se o número de policiais for dado como entrada, então o problema é PSPACE-completo utilizando uma redução do problema de formula booleana quantificada (MAMINO, 2013).

Além da complexidade de determinar o *cop-number*, um dos tópicos mais estudados sobre o tema é limitar seu valor por uma função do número de vértices do grafo. Em 1985, H. Meyniel conjecturou que $c(G) = O(\sqrt{n})$ para todo grafo G com n vértices. A conjectura de Meyniel permanece como um dos maiores desafios no estudo do *cop-number*. O melhor limite conhecido foi obtido por Lu e Peng em 2012 (LU and PENG, 2012) e estabelece que $c(G) = O\left(\frac{n}{2^{1-o(n)}\sqrt{\log_2 n}}\right)$.

Embora a conjectura de Meyniel continue sem solução, podemos abordá-la para classes de grafos restritas, visando encontrar um limite para o *cop-number* de tais classes. Aigner e Fromme mostraram que todo grafo planar possui *cop-number* no máximo 3 (AIGNER and FROMME, 1984). Andreae generalizou o resultado de Aigner e Fromme para grafos que não possuem K_5 como menor ou que não possuem $K_{3,3}$ como menor (ANDREAE, 1984) (os grafos planares são exatamente aqueles que não possuem nem K_5 nem $K_{3,3}$ como menor). Além disso, Andreae ainda provou que para qualquer grafo H , o *cop-number* da classe de grafos livres de menor H é limitado por uma constante. Em 2012, Lu e Peng mostraram que se G é um grafo com diâmetro 2, ou um grafo bipartido com diâmetro no máximo 3, então $c(G) \leq 2\sqrt{n} - 1$ (LU and PENG, 2012). Mencionamos que Baird e Bonato publicaram um levantamento (BAIRD and BONATO, 2012) no qual relatam os principais resultados relacionados a conjectura de Meyniel.

Em 2009, Bonato et al. introduziram um novo parâmetro relacionado ao jogo de polícia e ladrão, o *k-capture time* (BONATO *et al.*, 2009). O *k-capture time* de um grafo G é o número mínimo de turnos do jogo de polícia e ladrão em G que k policiais precisam para capturar um ladrão, assumindo que este tentará adiar a captura o máximo possível. Bonato et al. mostraram que, embora calcular o *capture time* seja polinomial quando o número de policiais é fixo, o problema se torna NP-completo para um número arbitrário de policiais.

Em 2011, Bonato e Nowakowski publicaram um livro com os principais resultados sobre o jogo de polícia e ladrão (BONATO and NOWAKOWSKI, 2011), incluindo resultados sobre os diversos jogos derivados do mesmo.

No Capítulo 4, mostramos valores exatos para o *cop-number* e o *k-capture time* de grafos P_4 -tidy e $(q, q - 4)$ -grafos, além de um algoritmo polinomial para calculá-los. Obtemos este resultado utilizando uma decomposição dos grafos P_4 -tidy, introduzida por Giakoumakis (GIAKOUMAKIS, ROUSSEL, and THUILLIER, 1997), e a decomposição primeval dos $(q, q - 4)$ -grafos, introduzida por Babel e Olariu (BABEL and OLARIU, 1998). Além disso, mostramos que a Conjectura de Meyniel é válida para grafos P_4 -tidy conexos e para $(q, q - 4)$ -grafos conexos com pelo menos q vértices.

1.3 Jogo do Espião

Em 2016, introduzimos um novo jogo de perseguição em grafos, o *jogo do espião* (COHEN *et al.*, 2018). Diferentemente do que ocorre no jogo de polícia e ladrão, no jogo do espião o objetivo não é capturar o fugitivo, mas sim evitar que ele se distancie dos guardas o suficiente para se comunicar com seus comparsas.

Inicialmente um dos jogadores posiciona seu espião em um vértice de G e depois o outro jogador posiciona seus guardas. A partir de então os jogadores se revezam, começando pelo espião, a mover suas respectivas peças entre os vértices de G . O espião, em uma jogada, pode se mover para qualquer vértice a uma distância no máximo s (sua velocidade) de sua posição atual, enquanto cada policial só pode mover-se para vértices adjacentes, isto é cada policial tem velocidade 1. Assim como ocorre no Jogo de Polícia e Ladrão o Jogo do Espião transcorre com informação perfeita. O espião deseja se comunicar com seu país de origem e repassar os segredos que descobriu, no entanto ele só pode fazê-lo se, após a movimentação dos guardas, estiver a uma distância maior que d de qualquer vértice ocupado por um guarda. O espião vence se após um número finito de jogadas conseguir se comunicar com seu país, os guardas vencem se jamais permitem que o espião se comunique.

Novamente, em um grafo com n vértices, tem-se que n guardas são suficientes para impedir que um espião vença. Assim, desejamos obter o mínimo número de guardas para garantir que o espião jamais se comunique com seus companheiros. De modo similar, podemos desejar calcular, para um número k fixo, qual a máxima distância d tal que garantimos que um grupo de k guardas consegue manter, indefinidamente, o espião a uma distância no máximo d de pelo menos um dos guardas a cada turno.

Podemos definir ainda uma versão do jogo do espião na qual os guardas são posicionados primeiro. No entanto, o espião poderia ser posicionado à uma distância maior que d de todos os guardas já no primeiro turno. Neste caso, temos que mudar as condições de vitória dos jogadores para o seguinte: o jogador que controla os guardas vence se, após um número finito de rodadas, ele consegue manter pelo menos um guarda a distância no máximo d do espião a cada turno, caso contrário o jogador que controla o espião vence. Isto é, o espião precisa se comunicar indefinidamente para vencer, enquanto

os guardas precisam garantir que o espião se comunique no máximo um número finito de vezes.

Em 2010, Bonato et al. sugeriram uma generalização do jogo de polícia e ladrão na qual o ladrão era capturado se a qualquer momento ele estivesse a uma distância no máximo d de um policial (BONATO, CHINIFOROOSHAN, and PRALAT, 2010). Esta variação do jogo ficou conhecida como jogo de polícia e ladrão com raio de captura. Observe que a versão do jogo do espião em que os guardas são posicionados primeiro e o espião tem velocidade $s = 1$ é equivalente ao proposto por Bonato. Esta equivalência no entanto se desfaz quando $s > 1$. Podemos observar que, no jogo com raio de captura d , 1 policial é suficiente para prender um ladrão quando o grafo é uma árvore. No entanto, conforme mostraremos no Capítulo 5, normalmente 1 guarda não é suficiente para patrulhar nem mesmo um caminho no jogo do espião com $s > 1$. Isto se deve ao fato de guardas e espião poderem ocupar o mesmo vértice simultaneamente durante o curso do jogo.

No jogo de *dominação eterna*, um conjunto de k defensores se posicionam inicialmente no grafo. A cada turno, um atacante seleciona um vértice qualquer do grafo para ser atacado. Após isto, cada defensor escolhe entre se mover para um vértice adjacente ou permanecer no atual. O jogador atacante vence se em qualquer turno, após o movimento dos defensores, não há nenhum defensor a uma distância no máximo d (parâmetro fixo) do vértice atacado neste turno. Os defensores vencem caso contrário.

Basicamente, para que os defensores sejam vitoriosos, antes de cada ataque não pode existir vértice a uma distância superior a $d + 1$ de todos eles. O jogo do espião com distância d e velocidade s ilimitada (ou pelo menos igual ao diâmetro do grafo) é equivalente ao jogo de dominação eterna.

Os resultados que mostram a dificuldade computacional do problema descritos posteriormente neste texto, bem como a relação do problema com outros já estabelecidos na literatura, comprovam a importância teórica do jogo do espião.

No Capítulo 5, estudamos o problema do jogo do espião e mostramos que determinar o número mínimo de guardas para que o jogador “perseguidor” vença o jogo é NP-difícil para qualquer velocidade $s \geq 2$ e qualquer distância d e, além disso, $(1 - o(1)) \ln n$ -inaproximável em tempo polinomial a menos que $P = NP$. Também mostramos limites superiores para o número de guardas necessários quando o jogo ocorre em caminhos e ciclos e que a versão direcionada do jogo é PSPACE-completa quando o jogo ocorre em DAG's. Parte dos resultados deste capítulo foram publicados em (COHEN *et al.*, 2018).

2 CONCEITOS BÁSICOS

Neste capítulo iremos enunciar notações e conceitos estabelecidos na literatura que serão utilizados nos capítulos seguintes. Na seção 2.1 enunciamos o conceitos básicos relacionados a Grafos, na seção 2.2 falamos sobre grafos direcionados, nas seções 2.3 e 2.4 enunciamos alguns resultados estruturais sobre diversas classes de grafos. Por fim, nas seções 2.5 e 2.6, tratamos respectivamente de problemas com parâmetro fixo e inaproximabilidade.

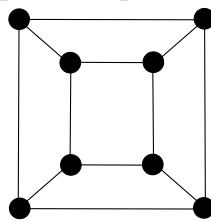
2.1 Grafos

Um *grafo* G é um par ordenado $(V(G), E(G))$ composto de um conjunto, não vazio, $V(G)$ de *vértices* e um conjunto $E(G)$, disjunto de $V(G)$, de *arestas*. Cada aresta é um par não ordenado de vértices de G . Se o par (u, v) está em $E(G)$, escreve-se $uv \in E(G)$ e diz-se que u e v são *extremidades* da aresta uv .

A denominação grafo vem do fato dele poder ser representado graficamente, os vértices como pontos e as arestas como linhas ligando suas extremidades. Dizemos que dois vértices são *adjacentes* quando eles são extremidades de uma mesma aresta. De maneira análoga duas arestas são adjacentes se compartilham uma extremidade. Uma aresta que possui extremidades iguais é chamada *laço*. Duas arestas distintas com extremidades iguais são chamadas de *arestas múltiplas*.

Um grafo G é *finito* se $V(G)$ e $E(G)$ são finitos. Um grafo G é dito *simples* se G é finito, não possui laços e não possui arestas múltiplas. A partir de agora, toda vez que mencionarmos grafo, estaremos nos referindo a um grafo simples.

Figura 2 – Exemplo de representação de um grafo



Fonte: Elaborado pelo autor.

O *complemento* de um grafo $G = (V, E)$ é o grafo $\overline{G} = (V, E')$ tal que $uv \in E'$ se e somente se $uv \notin E$. Um grafo que possui uma aresta entre qualquer par de vértices é chamado de *grafo completo*. Um grafo completo com n vértices é denotado por K_n . Um grafo que não possui nenhuma aresta é chamado de *grafo vazio*.

O *grau* de um vértice v em G é o número de vértices que são adjacentes a v em G e é denotado por $d_G(v)$. Se não houver ambiguidade com relação ao grafo a que nos referimos denota-se o grau de um vértice v apenas por $d(v)$. O *grau máximo* de um grafo G é o maior grau entre os vértices de G e é denotado por $\Delta(G)$.

Dado um um vértice v de um grafo G , $N(v)$ denota o conjunto dos vizinhos, ou vizinhança, de v . A vizinhança fechada de v , denotada por $N[v]$, é o conjunto $N(v) \cup \{v\}$. De maneira semelhante podemos pensar na vizinhança de um conjunto de vértices $X \subseteq V(G)$ como $N(X) = \bigcup_{v \in X} N(v)$.

Um *conjunto dominante* D de um grafo G é um subconjunto de $V(G)$ tal que, para todo $v \in V(G) \setminus D$ v , é adjacente a pelo menos um vértice de D . O *número de dominação* de um grafo G , denotado por $\gamma(G)$, é igual a cardinalidade de um conjunto dominante mínimo.

Seja $G = (V, E)$ um grafo. O grafo $H = (V', E')$ é dito um *subgrafo* de G se $V' \subseteq V$ e se $E' \subseteq E$. Seja $V' \subseteq V$, o subgrafo de G cujo conjunto de vértices é V' e o conjunto de arestas é o conjunto de todas as arestas de G que tem ambas as extremidades em V' é chamado de *subgrafo de G induzido por V'* e é denotado por $G[V']$. Podemos dizer também que $G[V']$ é um subgrafo induzido de G . Chamamos um conjunto de vértices que induzem um grafo completo de *clique* e um conjunto de vértices que induzem um grafo vazio de *conjunto independente*. O número de vértices da maior clique de G é o seu *número de clique* e é denotado por $\omega(G)$. Um grafo G é dito *split* se seu conjunto de vértices pode ser particionado em uma clique e um conjunto independente.

Um *trilha* é uma sequência não vazia $W = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k$ na qual os termos são alternadamente vértices e arestas sem repetição de arestas, tais que, para $1 \leq i \leq k$, $e_i = v_{i-1} v_i$. Neste caso v_0 e v_k são as *extremidades* de W , enquanto os demais vértices são chamados de *vértices internos* de W . Como trataremos apenas de grafos simples, podemos omitir as arestas na descrição de trilhas. Um *caminho* é uma trilha sem repetição de vértices.

Um caminho com n vértices é denotado por P_n e o *comprimento* de um caminho é o número de arestas que este possui. A *distância entre dois vértices u e v* em um grafo G é o comprimento do menor caminho que possui extremidades u e v em G . Um grafo G é dito *conexo* se para todo par $u, v \in V(G)$ existe um caminho com extremidades u e v em G , caso contrário G é *desconexo*.

Um *ciclo* é um trilha cujas extremidades são iguais e os demais vértices são todos distintos. Se um grafo G não possui nenhum ciclo como subgrafo então G é dito *acíclico*. Um grafo acíclico é chamado de *floresta*. Ademais, se o grafo for conexo é chamado de *árvore*.

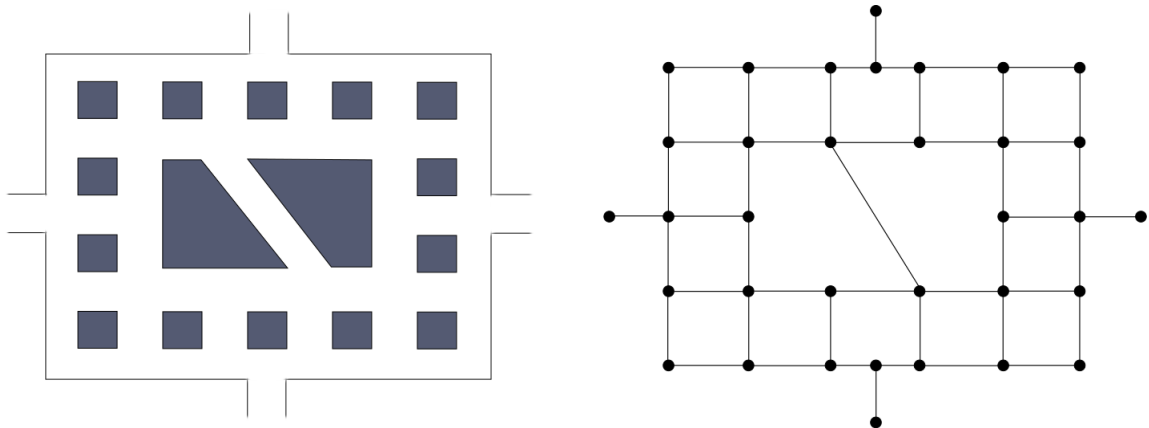
Seja G um grafo, uma *coloração* de G é uma função $c : V \rightarrow \mathbb{N}$. Uma coloração de um grafo G com k cores é chamada de *k -coloração* de G . Uma coloração de G é dita *própria* se para qualquer aresta uv de G , $c(u) \neq c(v)$. Uma coloração própria trivial pode ser obtida colorindo cada vértice do grafo com uma cor distinta dos demais. Assim no problema de coloração própria deseja-se obter o menor número de cores k para qual G pode ser k -colorido. O número cromático de um grafo G , $\chi(G)$, é o menor k tal que G admita um k -coloração própria. Dado um grafo G , decidir se $\chi(G) \leq k$ é um problema

NP-completo (KARP, 1972).

2.2 Grafos Direcionados

Diversos problemas computacionais podem ser modelados utilizando o conceito de grafo como descrito na seção 2.1. No entanto, para alguns problemas o conceito de grafo pode não ser o mais adequado. Imagine, por exemplo, o problema de um motorista tentando chegar a um determinado endereço em uma cidade. Poderíamos neste caso modelar o problema atribuindo um vértice a cada esquina e ligando por uma aresta dois vértices caso as esquinas por eles representadas compartilhem um quarteirão como mostrado abaixo.

Figura 3 – Mapa de uma cidade e sua representação em um grafo



Fonte: Elaborado pelo autor.

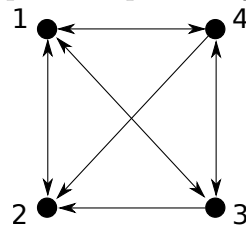
No entanto observe que a representação em grafo não nos fornece toda a informação necessária para resolver o problema do motorista, basta lembrar que algumas ruas podem ter apenas um sentido. Assim computar um caminho entre dois vértices do grafo não necessariamente nos fornece uma rota válida para o motorista.

Para resolver o problema do motorista devemos utilizar a noção de *grafo direcionado*. Neste caso cada aresta possui uma direção saindo de uma de suas extremidades e chegando à outra. Mais formalmente, Um *grafo direcionado* (ou *digrafo*) D é um par ordenado $(V(D), A(D))$ composto de um conjunto, não vazio, $V(D)$ de *vértices* e um conjunto $A(D)$, disjunto de $V(D)$, de *arcos*. Cada arco é um par ordenado de vértices de D , esta é a diferença entre grafos e digrafos. Se o par (u, v) está em $A(D)$ diz-se que o arco vai de u para v , além disso chamamos u de *cauda* e v de *cabeça* do arco.

Assim como no caso dos grafos, digrafos também possuem representação gráfica. Os vértices continuam sendo representados por pontos, enquanto os arcos são representados por setas saindo do vértice que representa sua cauda e chegando no vértice que representa sua cabeça. Ademais, se para algum par de vértices u e v os arcos (u, v) e (v, u) estão presentes no digrafo, representa-se ambos os arcos com uma única seta

bidirecional.

Figura 4 – Exemplo de representação de um digrafo



Fonte: Elaborado pelo autor.

Uma *trilha direcionada* em um digrafo D é uma sequência não vazia de vértices e arcos de D , sem repetição de arcos, $W = v_0, a_1, v_1, a_2, v_2 \dots, a_k, v_k$ tal que para todo $i = 1, \dots, k$, temos que $a_i = (v_{i-1}, v_i)$. Um *caminho direcionado* em um digrafo D é uma trilha direcionada sem repetição de vértices. Sempre que nos referimos a um caminho em um digrafo estamos nos referindo a um caminho direcionado, a não ser que seja explicitamente dito o contrário. Assim, o problema do motorista pode ser resolvido ao encontrar um caminho no digrafo que representa as ruas da cidade com suas respectivas orientações.

Um *ciclo* em um digrafo D é uma trilha direcionada cujas extremidades são iguais e os demais vértices são todos distintos. Se um digrafo D não possui nenhum ciclo direcionado como subgrafo então D é um *DAG* (*Directed Acyclic Digraph*).

Assim como no caso dos grafos os vértices de um digrafo também possuem graus, neste caso cada vértice v possui dois tipos de graus, o *grau de entrada*, $d^-(v)$ e o *grau de saída* $d^+(v)$. O grau de entrada de um vértice v é o número de arcos que tem v como cabeça, enquanto o grau de saída é o número de arcos que possuem v como cauda. A *vizinhança de entrada* de um vértice v em um digrafo D é o conjunto $N^-(v) = \{x \in V(D) | (x, v) \in A(D)\}$, enquanto a *vizinhança de saída* é o conjunto $N^+(v) = \{x \in V(D) | (v, x) \in A(D)\}$.

2.3 Grafos com poucos P_4 's

Todos os problemas que estudaremos neste texto são NP-difíceis para grafos em geral. Desta maneira, é pouco provável que existam algoritmos polinomiais para resolvê-los. Um meio conveniente de contornar esta intratabilidade é solucionar o problema para classes específicas de grafos, uma vez que desta maneira podemos nos aproveitar de certas propriedades que não são válidas para grafos em geral.

Nesta seção iremos enunciar resultados importantes sobre grafos que possuem restrições em relação a quantidade de P_4 's induzidos presentes em sua estrutura. Limitar o número de P_4 's induzidos nos ajuda a encontrar importantes características estruturais destes tipos de grafo.

Dados grafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2, E_2)$, a *união disjunta* de G_1 e G_2 é o grafo $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ e a *junção* de G_1 e G_2 é o grafo $G_1 \vee G_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{uv : u \in V_1, v \in V_2\})$.

Um *cografo* é um grafo sem P_4 's induzidos (CORNEIL, LERCHS, and STEWART, 1981). Como o complemento de um P_4 é um P_4 , todo complemento de um cografo também é um cografo. Ademais, se G é um cografo, então todo subgrafo de G também o é. Portanto, os cografos tem características bem definidas (CORNEIL, PERL, and STEWART, 1985):

- K_1 é um cografo;
- Se G é um cografo, então \overline{G} também é um cografo;
- Se G e H são ambos cografos, então $G \cup H$ é um cografo.

Utilizando a caracterização acima obtemos uma decomposição dos cografos: A cada cografo G está associada uma única árvore binária enraizada T_G , chamada de co-árvore de G , cujas folhas são os vértices de G , enquanto os nós internos são de dois tipos: união (\cup) e junção (\vee). Estas operações são aplicadas sobre os dois filhos de cada nó interno. Isto é, se G é um cografo, então uma das seguintes propriedades é válida (CORNEIL, PERL, and STEWART, 1985):

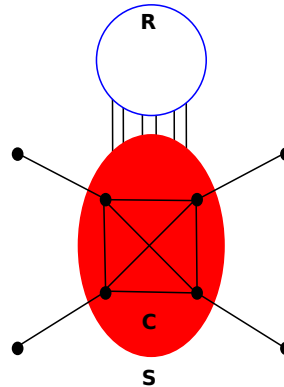
- G possui pelo menos um vértice;
- $G = G_1 \cup G_2$ é a união disjunta de dois cografos G_1 e G_2 .
- $G = G_1 \vee G_2$ é a junção de dois cografos G_1 e G_2 .

Um grafo é uma *aranha* (R, C, S) se o seu conjunto de vértices pode ser particionado em conjuntos R , C e S , onde $C = \{c_1, \dots, c_k\}$ e $S = \{s_1, \dots, s_k\}$ com $k \geq 1$ tais que:

- i. C é uma clique;
- ii. S é um conjunto independente;
- iii. Todo vértice de R é adjacente a todos os vértices de C e não-adjacente a todos os vértices de S ;
- iv. (a) s_i é adjacente a c_j se e só se $i = j$, para todos $1 \leq i, j \leq k$ (*aranha magra*); ou
(b) s_i é adjacente a c_j se e só se $i \neq j$, para todos $1 \leq i, j \leq k$ (*aranha gorda*)

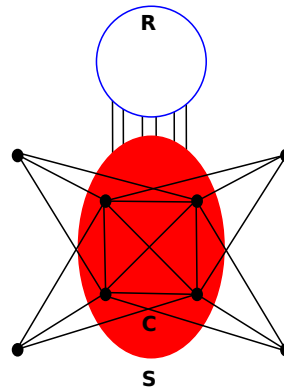
Chamamos R , C e S , respectivamente, de cabeça, corpo e patas da aranha. Note que R pode ser vazio e, nesse caso, dizemos que a aranha é sem cabeça.

Figura 5 – Aranha Magra



Fonte: Elaborado pelo autor.

Figura 6 – Aranha Gorda



Fonte: Elaborado pelo autor.

Um grafo G é P_4 -esparso se todo conjunto de 5 vértices em G induz no máximo um P_4 (JAMISON and OLARIU, 1992). Grafos P_4 -esparsos também possuem uma decomposição com operações de junção, união e formação de aranhas.

Teorema 2.1 (Caracterização dos P_4 -esparsos (JAMISON and OLARIU, 1992))

Um grafo G é P_4 -esparso se, e somente se, exatamente uma das condições abaixo é satisfeita:

- G possui no máximo um vértice;
- $G = G_1 \cup G_2$ é a união de dois grafos P_4 -esparsos G_1 e G_2 ;
- $G = G_1 \vee G_2$ é a junção de dois grafos P_4 -esparsos G_1 e G_2 ;
- G é uma aranha com partição (R, C, S) tal que $G[R]$ é um grafo P_4 -esparso.

Babel e Olariu definiram um grafo como $(q, q - 4)$ -grafo se nenhum conjunto com no máximo q vértices induz mais do que $(q - 4)$ P_4 's distintos (BABEL and OLARIU, 1998). Os cografos são exatamente os $(4, 0)$ -grafos (CORNEIL, LERCHS, and STEWART, 1981) e os grafos P_4 -esparsos são exatamente os $(5, 1)$ -grafos (HOÀNG, 1985).

Em 1998, Babel e Olariu provaram um importante resultado estrutural sobre os $(q, q - 4)$ -grafos. Este resultado caracteriza uma decomposição de tais grafos, chamada de *decomposição primeval* (BABEL and OLARIU, 1998), que pode ser obtida em tempo

linear $O(m + n)$ (BABEL *et al.*, 2001; BAUMANN, 1996), onde m e n são o número de arestas e de vértices do grafo, respectivamente.

Um grafo é *p-conexo* se, para toda partição dos vértices de G em conjuntos A e B não vazios, existe um P_4 com vértices de A e B . Uma *p-componente separável* é um subgrafo p -conexo maximal com uma partição (H_1, H_2) tal que todo P_4 induzido $wxyz$ é tal que $x, y \in H_1$ e $w, z \in H_2$.

Para definir esta decomposição, temos de definir primeiro algumas operações. Se G' é um grafo arbitrário e H é uma *p-componente separável* com separação (H_1, H_2) ou uma aranha sem cabeça com partição (\emptyset, H_1, H_2) , então a operação $G' \uplus H$ faz com que todos os vértices de G' sejam adjacentes a todos os vértices de H_1 e não adjacentes a todos os vértices de H_2 .

Teorema 2.2 (Decomposição Primeval (BABEL and OLARIU, 1998)) *Seja $q \geq 4$ um inteiro fixo e G um $(q, q - 4)$ -grafo, então G satisfaz um dos itens a seguir:*

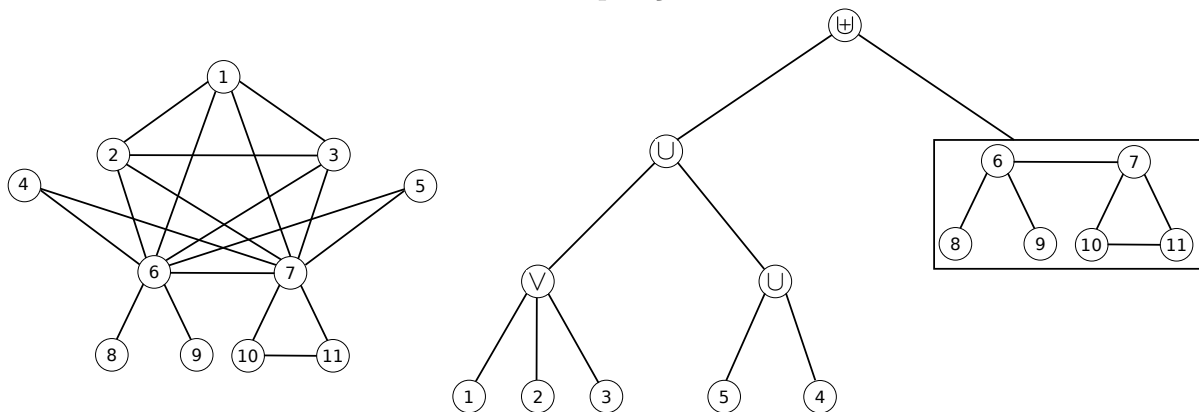
- (a) $G = G' \cup G''$;
- (b) $G = G' \vee G''$;
- (c) $G = G' \uplus H$;
- (d) G possui menos que q vértices.

onde G' e G'' são $(q, q - 4)$ -grafos e H é uma aranha (R, C, S) com $R = \emptyset$, ou uma p -componente separável com menos que q vértices.

É válido observar que todo grafo é $(q, q - 4)$ para algum valor de q .

O Teorema 2.2 leva a uma decomposição dos $(q, q - 4)$ -grafos. Seja T_G uma árvore de decomposição de um $(q, q - 4)$ -grafo G . Todo nó u de T_G representa um subgrafo $G(u)$ de G . A raiz r da árvore representa o próprio grafo G . As folhas de T_G são p -componentes com menos do que q vértices ou grafos aranha sem cabeça. Os nós internos são o resultado de operações (realizadas sobre os filhos de cada nó) de União disjunta ou Junção, que correspondem respectivamente, aos itens (a) e (b) do Teorema, ou uma operação que adiciona todas as arestas entre um grafo e uma parte determinada de outro grafo, correspondendo ao item (c).

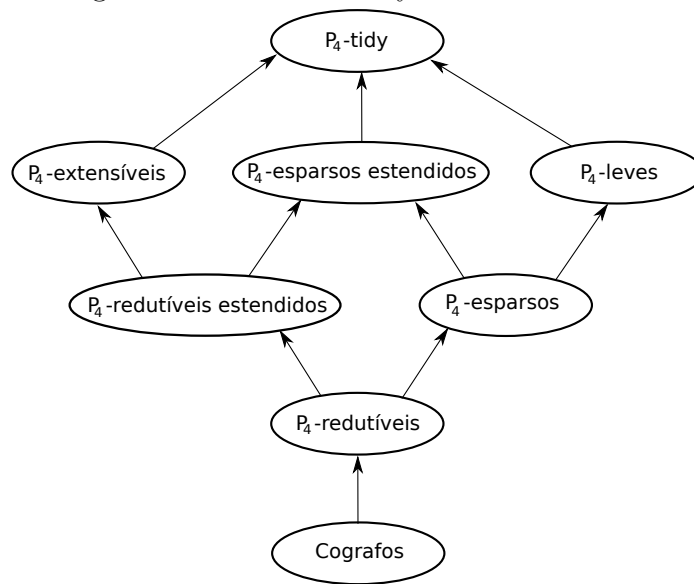
Figura 7 – Exemplo de um grafo e sua respectiva árvore de decomposição



Fonte: Elaborado pelo autor.

Um grafo G é P_4 -tidy se para todo P_4 induzido H em G , existe no máximo um vértice fora de H que induz um P_4 com três vértices de H . Pela definição podemos observar que todo grafo P_4 -esparso é também P_4 -tidy. De fato, a classe dos grafos P_4 -tidy inclui estritamente a classe dos grafos P_4 -esparcos. O C_5 , por exemplo, é um grafo P_4 -tidy que não pertence a classe dos P_4 -esparcos. Abaixo podemos ver algumas outras classes de grafos que estão contidas nos grafos P_4 -tidy.

Os grafos P_4 -tidy possuem uma decomposição bastante semelhante aos grafos estudados anteriormente (GIAKOUMAKIS, ROUSSEL, and THUILLIER, 1997). No entanto antes de abordarmos tal decomposição é necessário definirmos um outro tipo de grafo.

Figura 8 – Grafos P_4 -tidy e suas subclasses

Fonte: Elaborado pelo autor.

Uma *quase-aranha* é um grafo obtido de uma aranha (R, C, S) com no máximo um vértice de $S \cup C$ substituído por um grafo K_2 ou $\overline{K_2}$ (respeitando-se as adjacências). Pela definição, temos que toda aranha é também uma quase-aranha, onde nenhum vértice de $S \cup C$ foi substituído.

Após definirmos grafos quase-aranhas podemos avançar para a decomposição dos grafos P_4 -tidy.

Teorema 2.3 (GIAKOUMAKIS, ROUSSEL, and THUILLIER (1997)) *Um grafo G é P_4 -tidy se, e somente se, exatamente uma das condições abaixo é satisfeita:*

- (a) G é a união disjunta de dois grafos P_4 -tidy;
- (b) G é a junção de dois grafos P_4 -tidy;
- (c) G é uma quase-aranha, tal que a cabeça induz um grafo P_4 -tidy ou é vazia;
- (d) G é isomorfo a C_5 , $\overline{P_5}$ ou P_5 ;
- (e) $V(G)$ tem apenas um elemento.

De maneira similar aos anteriores, o Teorema 2.3 sugere uma decomposição em árvore dos grafos P_4 -tidy. Seja T_G uma árvore de decomposição de um grafo P_4 -tidy G . Todo nó u de T_G representa um subgrafo $G(u)$ de G . A raiz r da árvore representa o próprio grafo G . As folhas de T_G são K_1 , C_5 , P_5 , $\overline{P_5}$ ou grafos quase-aranhas sem cabeça. Os nós internos são o resultado de operações realizadas sobre os grafos de seus filhos. Dado um grafo G P_4 -tidy, com n vértices e m arestas, T_G é única, desconsiderando isomorfismos, e pode ser obtida em tempo $O(m + n)$ (GIAKOUMAKIS, ROUSSEL, and THUILLIER, 1997).

2.4 Grafos com Largura em Árvore Limitada

Por conta da estrutura de grafos do tipo árvore, diversos problemas intratáveis (NP-difíceis) podem ser resolvidos em tempo polinomial quando restritos a este tipo de grafos.

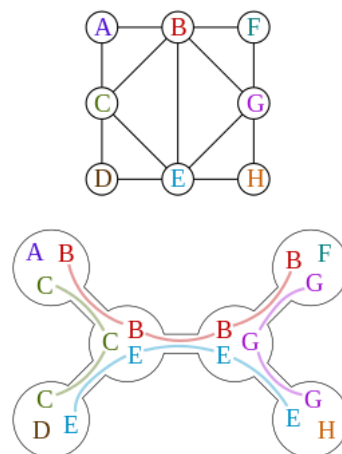
O conceito de largura em árvore foi introduzido por Robertson e Seymour (ROBERTSON and SEYMOUR, 1986). A largura em árvore tem por objetivo medir o quão similar um grafo é de uma árvore.

Antes de definirmos formalmente o conceito de largura em árvore, temos que definir o conceito de decomposição em árvore. Uma *decomposição em árvore* D de um grafo $G = (V, E)$ é um par (\mathcal{X}, T) onde $T = (I, F)$ é uma árvore e $\mathcal{X} = \{X_i | i \in I\}$ é uma família de subconjuntos de V , um para cada nó de T , tais que:

1. $\bigcup_{i \in I} X_i = V$;
2. Para toda aresta $uv \in E(G)$ existe $i \in I$ tal que $u \in X_i$ e $v \in X_i$;
3. Para todo $i, k, j \in I$ se k está no (único) caminho entre i e j em T , então $X_i \cap X_j \subseteq X_k$.

As duas primeiras propriedades garantem que todos os vértices e todas as arestas aparecem pelo menos uma vez na decomposição, enquanto a terceira propriedade garante a estrutura em árvore da decomposição. De fato, a terceira propriedade poderia ser reescrita da seguinte forma: Para todo $v \in V(G)$, $T_v = \{i \in I | v \in X_i\}$ induz uma subárvore de T .

Figura 9 – Exemplo de Decomposição em Árvore



Fonte: Wikipédia.

Um grafo pode admitir diversas decomposições em árvores possíveis. Por exemplo todo grafo G admite uma decomposição em árvore *trivial* $D = (\mathcal{X} = \{X_1 = V(G)\}, T = (\{1\}, \emptyset))$. Obviamente tal decomposição não nos diz praticamente nada sobre a semelhança do grafo G com uma árvore. Na verdade, quanto menor a cardinalidade dos elementos de \mathcal{X} mais semelhante G será de uma árvore. Árvores por exemplo admitem

decomposições nas quais para todo $i \in I$ temos $|X_i| = 2$.

A largura em árvore de uma decomposição $D = (\mathcal{X} = \{X_i | i \in I\}, T = (I, F))$ é $tw(D) = \max_{i \in I} \{|X_i| - 1\}$. A largura em árvore de um grafo G é a largura mínima dentre todas as decomposições em árvore possíveis de G . Em 1996, Bodlaender mostrou o seguinte resultado:

Teorema 2.4 (BODLAENDER, 1996) *Para todo $k \in \mathbb{N}$, existe um algoritmo de tempo linear que determina se um dado grafo G possui largura em árvore no máximo k e além disso calcula, se possível, uma decomposição em árvore de G com largura no máximo k .*

Uma decomposição em árvore $D = (\mathcal{X} = \{X_i | i \in I\}, T = (I, F))$ de um grafo $G = (V, E)$ é boa se a árvore T é binária, enraizada e seus nós são de quatro tipos:

Folha: Um nó i de T é do tipo folha, se ele não possui nós filhos em T e $|X_i| = 1$.

Introdução: Um nó i de T deste tipo possui um nó filho j e $X_i = X_j \cup v$ para algum $v \in V$.

Saída: Um nó i de T deste tipo possui um nó filho j e $X_i = X_j \setminus v$ para algum $v \in V$.

Junção: Um nó i de T deste tipo possui dois nós filhos j, l e $X_i = X_j = X_l$.

É interessante observar que dada uma decomposição em árvore, podemos convertê-la em tempo linear para uma decomposição em árvore boa (KLOKS, 1994). Esta decomposição, por se tratar de uma árvore binária com tipos específicos de nós, facilita o desenvolvimento de algoritmos para grafos com largura em árvore limitada.

De fato, Bodlaender descreveu uma técnica para resolver problemas em grafos, por exemplo o de coloração própria de vértices, utilizando a decomposição em árvore boa (BODLAENDER, 1997). Nesta abordagem, a decomposição é utilizada juntamente com técnicas de programação dinâmica para obter soluções dos problemas.

No Capítulo 3, utilizaremos boas árvores de decomposição para construir um algoritmo para o problema de lid-coloração para grafos com largura em árvore limitada.

2.5 Tratabilidade com Parâmetro Fixo

Se assumirmos que $P \neq NP$ diversos problemas bastante estudados na literatura só poderiam ser resolvidos em tempo de execução superpolinomial em relação ao tamanho da entrada fornecida no problema. No entanto diversos desses problemas podem ser resolvidos em tempo polinomial no tamanho da entrada e superpolinomial em um determinado parâmetro \mathcal{K} . Desta maneira se considerarmos apenas instâncias nas quais \mathcal{K} é limitado por um valor fixo, então o problema seria tratável, apesar de sua dificuldade no caso geral.

Um problema \mathcal{P} é dito *tratável com parâmetro fixo* (*fixed parameter tractable*, FPT) com parâmetro \mathcal{K} se dada uma entrada x do problema tal que $\mathcal{K}(x) = k$, é possível obter uma solução para o problema em tempo $f(k)n^c$, onde f é uma função que depende apenas de k , c é uma constante e n é o tamanho da entrada. Um algoritmo que resolve

um problema com tal tempo de execução é chamado de *algoritmo FPT* com parâmetro \mathcal{K} .

Dado um grafo G e um inteiro k , o problema de clique máxima é determinar se o grafo G possui uma clique com cardinalidade pelo menos k . O problema de clique máxima estava entre os famosos 21 problemas NP-completos de Karp (KARP, 1972).

Um maneira ingênua de resolver tal problema seria testar todos os subconjuntos de vértices de tamanho k e verificar se algum deles é uma clique. Observe que tal abordagem gera um algoritmo que possui tempo de execução $O(n^k)$, impraticável se k é parte da entrada. No entanto, se considerarmos k um inteiro fixo temos um algoritmo polinomial para o problema parametrizado em k . Ainda assim, não podemos caracterizá-lo como um problema FPT.

Se considerarmos o grau máximo de G , Δ , como um parâmetro fixo, no entanto, podemos obter um algoritmo FPT para o problema: Seja v um vértice de G , uma clique contendo v deve conter apenas vizinhos do próprio, assim enumeramos os 2^Δ subconjuntos de vizinhos de v e verificamos quais deles formam uma clique. Fazendo isso para cada vértice de G iremos, ao final do processo obter a maior clique de G , em tempo $O(2^\Delta \Delta^2 n)$.

Infelizmente nem todos os problemas parametrizados admitem algoritmos FPT. Um exemplo é o problema de k -coloração própria de vértices. Mesmo se considerarmos k um parâmetro fixo do problema, um algoritmo FPT para o problema parametrizado implicaria um algoritmo polinomial para o problema de 3-coloração, o que é improvável devido ao fato deste ser NP-completo.

2.6 Algoritmos Aproximativos e Inaproximabilidade

Um problema de otimização de minimização P pode ser definido como uma tripla (I_P, f_P, val_P) , onde I_P é o conjunto de instâncias do problema, f_P é uma função que relaciona a cada instância de $i \in I_P$ um conjunto $f_P(i)$ de soluções viáveis e val_P é uma função que, para cada par $i \in I_P$ e $s \in f_P(i)$, atribui um valor $\text{val}_P(i, s)$ (geralmente um real positivo) chamado de valor da solução. Dada uma instância $i \in I_P$ o objetivo do problema é encontrar uma solução viável $s^* \in f_P(i)$ que possua valor $\text{val}_P(i, s^*)$ mínimo dentre todas as soluções viáveis. Assim s^* é chamada de solução ótima desta instância. Problemas de otimização de maximização são definidos de maneira similar com a diferença que o objetivo é encontrar uma solução viável s^* com valor $\text{val}_P(i, s^*)$ máximo dentre todas as soluções viáveis.

Por exemplo, na versão de otimização do problema de coloração própria (CP), I_{CP} é o conjunto de todos os grafos, para cada grafo G , $f_{CP}(G)$ é o conjunto de todas as colorações próprias de G e, para cada coloração c dentre estas, $\text{val}_{CP}(G, c)$ é o número de cores utilizado em c . Como o problema de coloração própria é um problema de minimização o objetivo é encontrar a coloração c^* de G que utiliza o mínimo número de cores

dentre todas as colorações próprias de G .

Uma forma de estudar problemas NP-difíceis sem restringir as entradas fornecidas é tentar encontrar algoritmos polinomiais que, embora não achem uma solução ótima para toda instância fornecida, encontram uma solução viável cujo valor é no máximo $r > 1$ vezes pior que o valor de uma solução ótima. Um algoritmo com essa característica é chamado de r -aproximativo. Iremos explicar de maneira mais formal esse conceito nos parágrafos a seguir.

Dado um problema de otimização P , denotamos o valor da solução ótima para uma instância $i \in I$ de P por $\text{opt}_P(i)$. Seja ainda uma solução $s \in f_P(i)$, a *razão de performance* $\mathcal{R}_P(i, s)$ é definida por:

$$\mathcal{R}_P(i, s) = \max \left\{ \frac{\text{opt}_P(i)}{\text{val}_P(i, s)}, \frac{\text{val}_P(i, s)}{\text{opt}_P(i)} \right\}.$$

A razão de performance de um problema P nos diz o quão próximo o valor de uma solução viável está do valor de uma solução ótima de uma determinada entrada de P , independente do problema ser de minimização ou maximização.

Dada uma constante $r \geq 1$, um *algoritmo r -aproximativo* para P é um algoritmo que, aplicado para qualquer instância i de P , executa em tempo polinomial em relação ao tamanho de i e produz uma solução viável s de i tal que $\mathcal{R}_P(i, s) \leq r$. Se tal algoritmo existe para uma constante r , então P pertence à classe APX.

No entanto, nem todo problema de otimização NP-difícil pode ser aproximado para qualquer $r > 1$. Dizemos que um problema P é *r -inaproximável em tempo polinomial* se não existe um algoritmo polinomial r -aproximativo para P . Dada uma função $r(n)$, dizemos que o problema P é *$O(r(n))$ -inaproximável em tempo polinomial* se existe uma função $r'(n) = O(r(n))$ tal que o problema P é $r'(n)$ -inaproximável em tempo polinomial. O problema de coloração própria, por exemplo, é $O(n^{1-\varepsilon})$ -inaproximável para qualquer $\varepsilon > 0$ a menos que $P = NP$ (ZUCKERMAN, 2006).

3 COLORAÇÃO LOCALMENTE IDENTIFICÁVEL

Neste capítulo iremos abordar o problema de coloração localmente identificável e uma variação deste, conhecida como coloração localmente identificável forte. Iremos examinar a complexidade desses problemas apresentando tanto resultados que demonstram a dificuldade em solucioná-los como algoritmos polinomiais para resolvê-los quando as entradas obedecem a certas restrições.

3.1 Introdução

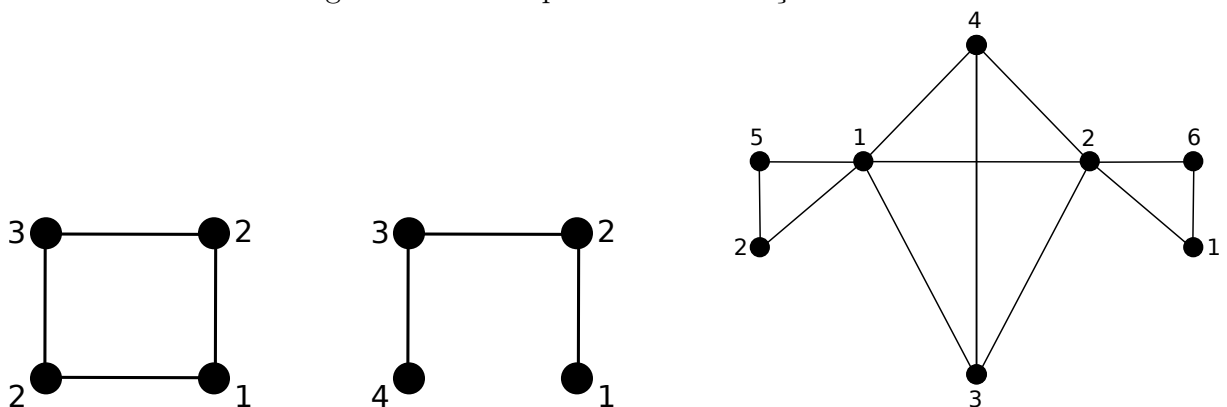
Dada uma coloração c dos vértices de um grafo G e um subconjunto $S \subseteq V(G)$, seja $c(S)$ o conjunto de cores dos vértices de S em c . Dado um subgrafo H de G e uma coloração c de G , podemos ainda escrever $c(H)$ para nos referir ao conjunto de cores dos vértices de H em c .

Uma coloração c é dita uma *coloração localmente identificável* (lid-coloração) de G se:

- (i) c é uma coloração própria dos vértices de G ;
- (ii) $\forall uv \in E(G)$, se $N[u] \neq N[v]$, então $c(N[u]) \neq c(N[v])$.

Podemos interpretar que, para todo vértice u , as cores de c em $c(N[u])$ formam um código que caracteriza u e o distingue dos seus vizinhos. Diversos problemas em teoria dos grafos tem como objetivo diferenciar elementos do grafo (vértices e arestas). O problema de códigos de identificação (KARPOVSKY, CHAKRABARTY, and LEVITIN, 1998), por exemplo, utiliza um conjunto dominante do próprio grafo para identificar cada vértice pelos vértices que o dominam.

Figura 10 – Exemplos de lid-colorações ótimas



Fonte: Elaborado pelo autor.

Obviamente, se colorirmos todos os vértices de um grafo G com cores distintas, teremos uma lid-coloração de G , uma vez que vértices adjacentes com vizinhanças fechadas distintas estariam “associados” a conjunto de cores distintos. Assim, o objetivo do problema de lid-coloração é determinar o menor número possível de cores para colorir

uma instância. O *número lid-cromático* de um grafo G , denotado por $\chi_{lid}(G)$, é o menor inteiro k tal que existe uma lid-coloração de G com k cores.

Colorações Localmente Identificáveis foram recentemente introduzidas por Esperet et al. (ESPERET *et al.*, 2012). Neste artigo eles mostraram que determinar se $\chi_{lid}(G) \leq k$ é NP-completo mesmo quando G é um grafo bipartido, mas é polinomial para árvores.

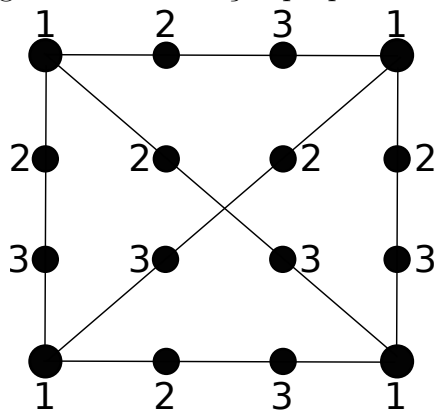
Esperet et al. mostraram ainda que $\chi_{lid}(G) \leq 2\omega(G) = 2\chi(G)$ se G é uma k -árvore ou grafo de intervalo e $\chi_{lid}(G) \leq 2\omega(G) = 2\chi(G) - 1$ se G é um grafo split ou cografo. Como k -árvores, grafos de intervalo e grafos split são subclasses de grafos cordais eles também conjecturaram o seguinte:

Conjectura 3.1 *Para qualquer grafo cordal G , $\chi_{lid}(G) \leq 2\chi(G)$*

Ninguém conseguiu provar ou refutar esta conjectura até o momento.

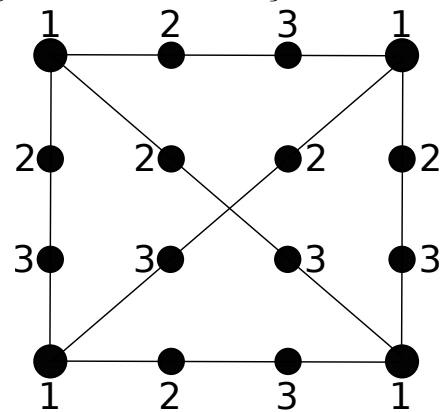
Embora $\chi_{lid}(G)$ seja limitado por uma função do número cromático para as classes acima, isto não é verdade para grafos em geral (ESPERET *et al.*, 2012). Tome por exemplo o grafo K_n^* obtido de uma clique com n vértices onde cada aresta é subdividida duas vezes. K_n^* possui uma coloração própria com 3 cores, basta colorir todos os vértices da clique original com a cor 1 e cada par de vértices gerado pela subdivisão de uma aresta com as cores 2 e 3. No entanto, $\chi_{lid}(K_n^*) \geq n$. Para constatar este limite basta observar que se dois vértices da clique original tiverem a mesma cor, os vértices gerados pela subdivisão da aresta entre eles terão o mesmo conjunto de cores em sua vizinhança fechada.

Figura 11 – Coloração própria de K_4^*



Fonte: Elaborado pelo autor.

Figura 12 – Lid-coloração ótima de K_4^*



Fonte: Elaborado pelo autor.

Iremos utilizar a idéia de subdivisão de arestas na Seção 3.3 para mostrar um resultado de inaproximabilidade de lid-coloração por meio de uma redução do problema de coloração própria.

Foucaud et al. mostraram que, dado um grafo G com grau máximo $\Delta \geq 3$ e uma lid-coloração c com mais que $2\Delta^2 - 3\Delta + 3$, podemos obter uma lid-coloração de G com no máximo $2\Delta^2 - 3\Delta + 3$ cores (FOUCAUD *et al.*, 2012), assim mostrando que

$\chi_{lid}(G) = O(\Delta^2)$, respondendo positivamente uma questão deixada por Esperet et al. (ESPERET *et al.*, 2012).

Esperet et al. ainda conjecturaram que todo grafo planar possui número lid-cromático limitado (ESPERET *et al.*, 2012). Gonçalves et al. mostraram que qualquer grafo planar possui uma lid-coloração com 1280 cores (GONÇALVES, PARREAU, and PINLOU, 2013), respondendo positivamente a esta conjectura.

Esperet et al. introduziram ainda o conceito de lid-coloração forte (ESPERET *et al.*, 2012). Dado um grafo G e uma lid-coloração c de G com cores $1, \dots, k$, diz-se que c é uma *lid-coloração forte* (ou *slid-coloração*) de G se, e somente se, para qualquer vértice v tal que $N[v] \neq V(G)$, tem-se que $c(N[v]) \neq \{1, \dots, k\}$. Isto é, todo vértice que “vê” todas as cores é um vértice universal. O menor inteiro k tal que G possui uma lid-coloração forte com k cores é o *número lid-cromático forte* (ou número *slid-cromático*) de G e é denotado por $\widetilde{\chi}_{lid}(G)$.

Por exemplo, no C_4 da Figura 10, temos uma lid-coloração ótima mas que não é forte, uma vez que os vértices com a cor 2 não são universais mas tem contato com todas as cores. Por outro lado, no último grafo da mesma figura mostramos uma lid-coloração forte.

Dado um cografo G , Esperet et al. mostraram que $\widetilde{\chi}_{lid}(G) \leq 2\omega(G)$ (ESPERET *et al.*, 2012). Foucaud et al. mostraram ainda que, dado um grafo G com grau máximo Δ , tem-se que $\widetilde{\chi}_{lid}(G) \leq 2\Delta^2 - \Delta + 1$.

Observe que, exceto pelos resultados para grafos bipartidos e árvores, todos os demais resultados obtidos para lid-coloração e slid-coloração são sobre limites para χ_{lid} e $\widetilde{\chi}_{lid}$. O objetivo deste capítulo é abordar a complexidade destes problemas, por exemplo obtendo algoritmos polinomiais para determinadas classes de grafos, e investigar os aspectos aproximativos dos mesmos. Ambas as questões foram pouco estudadas na literatura até então.

Na Seção 3.2, estudamos a relação entre $\widetilde{\chi}_{lid}$ e χ_{lid} . Mostramos que para qualquer grafo G , tem-se que $\widetilde{\chi}_{lid}(G) \leq \chi_{lid}(G) + 1$. Na Seção 3.3, provamos que $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ são $O(n^{1-\varepsilon})$ -inaproximáveis em tempo polinomial, para todo $\varepsilon > 0$, a menos que $P = NP$. Na Seção 3.4, mostramos um algoritmo linear para determinar $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ para $(q, q-4)$ -grafos. Finalmente, na Seção 3.5, mostramos um algoritmo polinomial para determinar $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ para grafos com largura em árvore limitada.

A maior parte dos resultados apresentados neste capítulo foram publicadas em (MARTINS and SAMPAIO, 2018)

3.2 Resultados Preliminares

Nesta seção obtemos resultados que relacionam o número lid-cromático e o número lid-cromático forte. De fato, na seção 3.4 iremos mostrar um algoritmo que utiliza

lid-colorações fortes para construir lid-colorações de grafos com poucos P_4 's. Tal resultado demonstra a importância de se estudar a relação entre estes dois parâmetros.

Lema 3.1 *Seja c uma lid-coloração de um grafo G com cores $\{1, \dots, k\}$ e $v \in V(G)$. A coloração c' obtida a partir de c mudando a cor do vértice v para $k+1$ é uma lid-coloração de G .*

Prova: Claramente a coloração c' é própria uma vez que v é o único vértice colorido com a cor $k+1$ e os demais vértices mantiveram as cores de c .

Observe que os únicos vértices que tiveram as cores de suas vizinhanças fechadas alteradas em c' foram o próprio v e seus vizinhos. Portanto, como a coloração c é uma lid-coloração, se existe uma aresta uw tal que $N[u] \neq N[w]$ e $c'(N[u]) = c'(N[w])$ temos que u ou w pertencem a $N[v]$. Ou seja, qualquer par de vértices que viola a condição (ii) em c' deve conter v ou um de seus vizinhos.

Admita que os vértices v e u violam a condição (ii). Observe que u não possui nenhum vizinho colorido com a cor $c(v)$ em c' , caso contrário $c'(N[u]) \neq c'(N[v])$, uma vez que $c(v) \notin c'(N[v])$. Observe também que $c'(N[v]) = (c(N[v]) \setminus \{c(v)\}) \cup \{k+1\}$ e que $c'(N[u]) = (c(N[u]) \setminus \{c(v)\}) \cup \{k+1\}$. Como $c'(N[v]) = c'(N[u])$ temos que $c(N[v]) = c(N[u])$. No entanto, como c é uma lid-coloração de G e $c(N[v]) = c(N[u])$ temos que $N[v] = N[u]$, uma contradição com o fato de u e v violarem a condição (ii).

Agora admita que os vértices u e w violam a condição (ii) e que u é vizinho de v . Como v é o único vértice colorido com a cor $k+1$, então w também tem que ser vizinho de v . Como $c'(N[u]) = c'(N[w])$, ou a cor $c(v)$ aparece em ambas as vizinhanças ou em nenhuma delas na coloração c' . Se a cor $c(v)$ aparece em ambas as vizinhanças em c' temos que $c'(N[u]) = c(N[u]) \cup \{k+1\}$ e $c'(N[w]) = c(N[w]) \cup \{k+1\}$. Se a cor $c(v)$ não aparece nas vizinhanças de u nem de w em c' então $c'(N[u]) = (c(N[u]) \setminus \{c(v)\}) \cup \{k+1\}$ e $c'(N[w]) = (c(N[w]) \setminus \{c(v)\}) \cup \{k+1\}$. Observe que em ambos os casos $c'(N[u]) = c'(N[w])$ se e somente se $c(N[u]) = c(N[w])$. Como c é uma lid-coloração temos que $N[u] = N[w]$, uma contradição com o fato de u e w violarem a condição (ii). \square

Dada uma coloração c dos vértices de um grafo G com cores $\{1, \dots, k\}$, dizemos que $v \in V(G)$ é um vértice *ruim* em relação a c se v não é um vértice universal e $c(N[v]) = \{1, \dots, k\}$. Observe que as slid-colorações de um grafo G são as lid-colorações que não possuem vértices ruins. Iremos utilizar o Lema 3.1 para mostrar o seguinte resultado.

Teorema 3.1 *Seja G um grafo qualquer. Então $\chi_{lid}(G) \leq \widetilde{\chi}_{lid}(G) \leq \chi_{lid}(G) + 1$.*

Prova: O limite $\chi_{lid}(G) \leq \widetilde{\chi}_{lid}(G)$ é trivial visto que toda slid-coloração de G é também uma lid-coloração.

Tome uma lid-coloração mínima c de um grafo G usando cores $\{1, \dots, \chi_{lid}(G)\}$. Se tal coloração não possui vértices ruins, então $\chi_{lid}(G) = \widetilde{\chi}_{lid}(G)$ e os limites são válidos. Então suponha que existe um vértice ruim $v \in V(G)$ em relação a c e, sem perda de generalidade, suponha que $c(v) = 1$. Pelo Lema 3.1 podemos mudar a cor de v para

$\chi_{lid}(G) + 1$ mantendo as propriedades de lid-coloração.

Caso exista outro vértice colorido com a cor 1 não fazemos mais alterações na coloração. Caso v fosse o único vértice da cor 1, como v não é universal, existe pelo menos um vértice w de G que não é vizinho de v . Como a cor 1 não está mais na coloração podemos usar novamente o Lema 3.1 e mudar a cor de w para 1 e mantermos a lid-coloração. Observe ainda que w não era o único vértice colorido com a cor $c(w)$, pois esta cor aparecia em um vértice na vizinhança de v . Desta maneira obtemos uma lid-coloração c' com cores $\{1, \dots, \chi_{lid}(G) + 1\}$ na qual nenhum vizinho de v é colorido com a cor 1, isto é, v não é um vértice ruim em relação a c' . Resta-nos mostrar que não existem outros vértices ruins na coloração c' .

Por contradição, suponha um vértice ruim $u \in V(G)$ em relação a coloração c' . Observe que u deve ser adjacente a v uma vez que este é o único vértice colorido com a cor $\chi_{lid}(G) + 1$. Além disso u deve ser adjacente a um vértice colorido com a cor 1 e, portanto, $N[u] \neq N[v]$. No entanto, podemos observar que se u é um vértice ruim em c' , então u também é um vértice ruim em c . Assim temos $c(N[u]) = \{1, \dots, \chi_{lid}(G)\} = c(N[v])$ e $N[u] \neq N[v]$. Uma vez que u e v são adjacentes, temos uma contradição com o fato da coloração original c ser uma lid-coloração. \square

Ainda sobre a relação entre estes parâmetros, podemos afirmar que, sempre que um grafo G possuir um vértice universal u , tem-se que $\chi_{lid}(G) = \widetilde{\chi}_{lid}(G)$. Isto se deve ao fato que, neste caso, qualquer lid-coloração c de G com k cores também será uma slid-coloração de G . Caso contrário, teríamos um vértice não universal v tal que $c(N[v]) = \{1, \dots, k\} = c(N[u])$. No entanto, como v não é universal temos que $N[v] \neq N[u]$, o que viola a condição (ii).

Entretanto, apenas a existência de um vértice universal não caracteriza todos os casos em que $\chi_{lid}(G) = \widetilde{\chi}_{lid}(G)$. Basta observar caso do P_4 (Figura 10) em que temos $\chi_{lid}(P_4) = \widetilde{\chi}_{lid}(P_4) = 4$ mesmo sem a existência de um vértice universal. Como exemplo de grafo no qual os dois parâmetros possuem valores distintos podemos citar o C_4 (Figura 10) em que temos $\chi_{lid}(C_4) = 3$ e $\widetilde{\chi}_{lid}(C_4) = 4$, ou seja o limite fornecido pelo Teorema 3.1 é apertado.

Seja G um grafo com grau máximo Δ . O Teorema 3.1 juntamente com o limite $\chi_{lid}(G) \leq 2\Delta^2 - 3\Delta + 3$ mostrado por Foucaud et al. (FOUCAUD *et al.*, 2012) implicam que $\widetilde{\chi}_{lid}(G) \leq 2\Delta^2 - 3\Delta + 4$ melhorando o limite anterior de $2\Delta^2 - \Delta + 1$, obtido também por Foucaud et al. no mesmo artigo.

3.3 Inaproximabilidade dos parâmetros de lid-coloração

Uma *redução* de um problema P_1 para um problema P_2 consiste de um par (f, g) de funções computáveis em tempo polinomial tais que, para qualquer instância I de P_1 , (a) $f(I)$ é uma instância de P_2 e (b) $g(I, S)$ é uma solução viável de I , para qualquer

solução viável S de $f(I)$.

Uma *redução contínua* de P_1 para P_2 é uma tripla (f, g, γ) , onde (f, g) é uma redução de P_1 para P_2 e $\gamma \geq 1$ é uma constante, tal que, se $\mathcal{R}_{P_2}(f(I), S) \leq r$ ($r \geq 1$), então $\mathcal{R}_{P_1}(I, g(S)) \leq \gamma r$ para qualquer instância I de P_1 e para toda solução viável S de $f(I)$. A partir dessa definição se existe um algoritmo r -aproximativo de tempo polinomial para P_2 para algum $r \geq 1$, então existe um algoritmo γr -aproximativo de tempo polinomial para P_1 . Conseqüentemente, se P_1 é $O(r(n))$ -inaproximável em tempo polinomial, então P_2 também é $O(r(n))$ -inaproximável em tempo polinomial, onde $r(n)$ é uma função tal que $\lim_{n \rightarrow \infty} r(n) = \infty$.

A seguir mostramos uma redução contínua do problema de coloração própria para os problema de lid-coloração e lid-coloração forte.

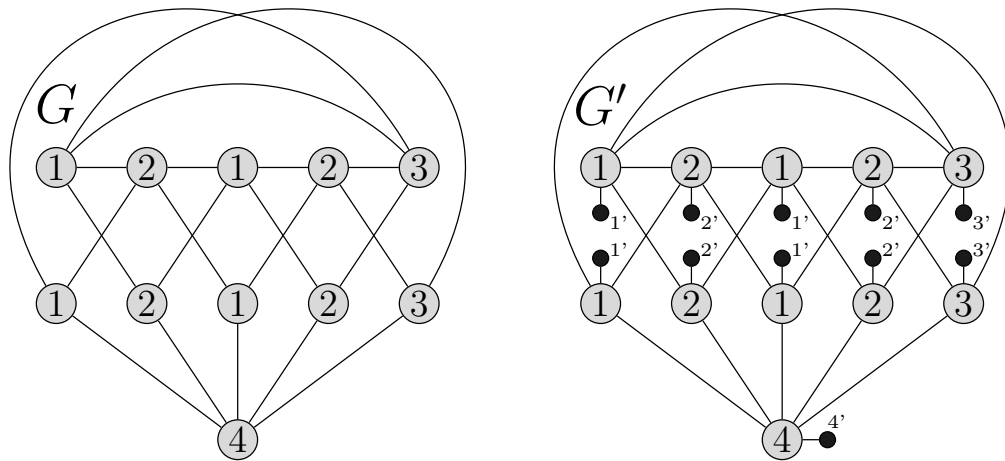
Teorema 3.2 *Determinar $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ são problemas $O(n^{1-\varepsilon})$ -inaproximáveis em tempo polinomial, para todo $\varepsilon > 0$, a menos que $P = NP$.*

Prova:

Nós obtemos uma redução contínua do problema de determinar $\chi(G)$.

Dado um grafo G sem vértices isolados, seja G' o grafo obtido de G ao adicionarmos um vértice v' adjacente a cada vértice v de G . Veja a Figura 13 para um exemplo com o grafo de Grötzsch, o menor grafo livre de triângulos com $\chi(G) = 4$.

Figura 13 – Exemplo de redução para o grafo de Grötzsch. Apresentamos uma coloração própria de G e uma lid-coloração de G'



Fonte: Elaborado pelo autor.

Observe que qualquer lid-coloração de G' induz uma coloração própria dos vértices de G . Portanto $\chi(G) \leq \chi_{lid}(G')$.

Ademais, para qualquer coloração própria c de G com k cores, podemos obter uma lid-coloração forte c' de G' utilizando $2k$ cores: se v tem cor i em G , colorimos v e v' respectivamente com as cores i e i' em G' . Note que c' é uma lid-coloração forte de G' , uma vez que, para qualquer aresta uv de G , $c(u)' \in c'(N[u])$ mas $c(u)' \notin c'(N[v])$ (pois $c(u) \neq c(v)$) e, para qualquer vértice v' adjacente somente a um vértice original v de G ,

$c'(N[v]) \neq c'(N[v'])$ (como v não é isolado, então $c(u) \in c'(N[v])$ mas $c(u) \notin c'(N[v'])$) para todo vizinho u de v em G). Portanto $\chi(G) \leq \chi_{lid}(G') \leq 2\chi(G)$.

Seja f a função que dado um grafo G retorna o grafo G' e g a função que dada uma lid-coloração c' de G' retorna a coloração própria c de G induzida por c' . Como ambas as funções são computáveis em tempo polinomial e $\chi(G) \leq \chi_{lid}(G') \leq 2\chi(G)$, então a tripla $(f, g, 2)$ é uma redução contínua de coloração própria para slid-coloração de grafos.

Sabe-se que, a menos que $P = NP$, não existe algoritmo de tempo polinomial que aproxima o número cromático de um grafo por um fator melhor que $O(n^{1-\varepsilon})$ para qualquer $\varepsilon > 0$ (ZUCKERMAN, 2006). Como $|V(G')| = 2 \cdot |V(G)|$, então $\chi_{lid}(G')$ é $n^{1-\varepsilon}$ -inaproximável em tempo polinomial, para qualquer $\varepsilon > 0$, a menos que $P = NP$. \square

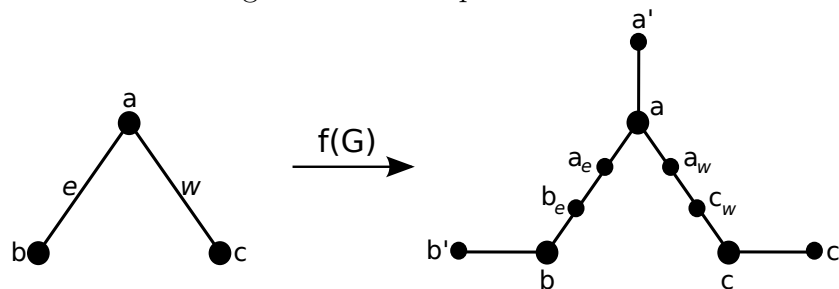
Podemos provar uma versão do teorema acima para grafos com cintura pelo menos 9. No entanto, o fator de inaproximabilidade na versão abaixo é pior que no Teorema 3.2.

Teorema 3.3 *Seja G um grafo com cintura pelo menos 9, determinar $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ são problemas $O(n^{\frac{1}{2}-\varepsilon})$ -inaproximáveis em tempo polinomial, para todo $\varepsilon > 0$, a menos que $P = NP$.*

Prova: Nós obtemos uma redução contínua do problema de determinar $\chi(G)$. Consideraremos instâncias G tais que $\chi(G) \geq 3$.

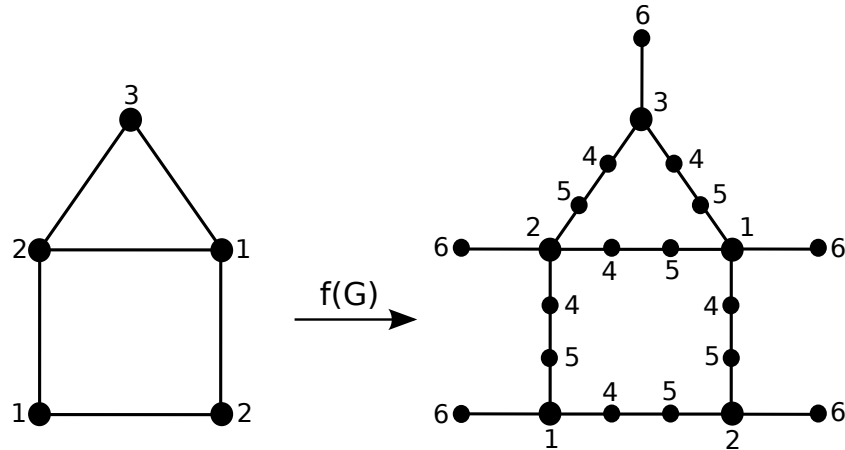
Dado um grafo G construímos o grafo G' da seguinte maneira. Iniciamos a construção com o conjunto de vértices do grafo original e o conjunto de arestas vazio. Para cada vértice $v \in V(G)$ adicionamos um vértice v' e fazemos v adjacente a v' em G' . Para cada aresta $e = uv \in E(G)$ adicionamos dois vértices u_e e v_e e fazemos u adjacente a u_e , u_e adjacente a v_e e v_e adjacente a v . Observe os exemplo das Figuras 14 e 15.

Figura 14 – Exemplo de G e G'



Fonte: Elaborado pelo autor.

Primeiro mostramos que qualquer lid-coloração ou slid-coloração de G' induz uma coloração própria dos vértices originais de G : Suponha uma (s)lid-coloração c de G' na qual um par de vértices $u, v \in V(G)$, adjacentes em G por uma aresta e , tenham cores iguais. Assim no grafo G' teríamos que $c(N[u_e]) = c(N[v_e]) = \{c(v), c(u_e), c(v_e)\}$. Como u_e e v_e são adjacentes e possuem vizinhanças distintas, a condição (II) é violada e,

Figura 15 – Exemplo de Coloração Própria de G e lid-Coloração de G' 

Fonte: Elaborado pelo autor.

portanto, c não seria (s)lid-coloração de G' . Dessa maneira temos que $\chi(G) \leq \chi_{lid}(G') \leq \widetilde{\chi}_{lid}(G')$.

Além disso, a partir de uma coloração própria mínima c de G com cores $\{1, \dots, \chi(G)\}$ é possível obter uma uma slid-coloração c' de G' utilizando $\chi(G) + 3$ cores. Para cada vértice v do grafo original fazemos $c'(v) = c(v)$. Para os vértices do tipo v' fazemos $c'(v') = \chi(G) + 1$ e para cada aresta $e = uv$ de G fazemos $c(u_e) = \chi(G) + 2$ e $c(v_e) = \chi(G) + 3$.

Iremos mostrar que c' é uma slid-coloração de G' . No grafo G' temos apenas três tipos de arestas: vv_e (um vértice do grafo original e um vértice gerado pela subdivisão de uma aresta $e = uv$ de G), vv' e u_ev_e (dois vértices gerados pela subdivisão da aresta $e = uv$ de G). Em nenhum destes casos as extremidades recebem a mesma cor em c' . Ademais pela construção de c' sabemos que, para qualquer vértice v do grafo original, tem-se que $c'(N[v])$ possui a cor $\chi(G) + 3$ e pelo menos uma dentre as cores $\chi(G) + 1$ e $\chi(G) + 2$, o que torna $c'(N[v])$ diferente tanto de $c'(N[v']) = \{\chi(G) + 3, c(v)\}$ quanto de $c'(N[v_e]) = \{\chi(G) + 1, \chi(G) + 2, c(v)\}$. Finalmente, para qualquer aresta $e = uv$ de G temos que $c'(N[v_e]) = \{\chi(G) + 1, \chi(G) + 2, c(v)\}$ e $c'(N[u_e]) = \{\chi(G) + 1, \chi(G) + 2, c(u)\}$, e como c é uma coloração própria de G , temos que $c'(N[u_e]) \neq c'(N[v_e])$.

Assim, c' é lid-coloração e, como nenhum vértice está em contato com todas as cores de c' , também é slid-coloração de G' . Portanto $\widetilde{\chi}_{lid}(G') \leq \chi(G) + 3 \leq 2\chi(G)$, uma vez que $\chi(G) \geq 3$.

Seja f a função que transforma G em G' e g a função que dada uma slid-coloração de G' obtém uma coloração própria de G . Como ambas as funções são computáveis em tempo polinomial e $\chi(G) \leq \chi_{lid}(G') \leq \widetilde{\chi}_{lid}(G') \leq \chi(G) + 3 \leq 2\chi(G)$ temos que a tripla $(f, g, 2)$ é uma redução contínua de Coloração Própria para slid-Coloração de grafos.

É conhecido que, a menos que $P = NP$, não existe algoritmo de tempo polinomial que aproxima o número cromático de um grafo por um fator melhor que $O(n^{1-\varepsilon})$ para

qualquer $\varepsilon > 0$ (ZUCKERMAN, 2006). Observe, no entanto, que se $G = (V, E)$, então $f(G)$ possui $O(|V|^2)$ vértices. Portanto, seja G um grafo com n vértices, os problemas de determinar $\widetilde{\chi}_{lid}(G)$ e $\chi_{lid}(G)$ são $O(n^{\frac{1}{2}-\varepsilon})$ -inaproximáveis, a menos que $P = NP$.

Resta-nos mostrar que o para qualquer grafo G , $f(G)$ possui cintura pelo menos 9. Para tanto basta observar que todo triângulo no grafo G é transformado em um C_9 em $f(G)$. \square

3.4 Lid-Colorações de Grafos com Poucos P4's

Nesta seção iremos utilizar a decomposição primeval dos $(q, q - 4)$ -grafos, como descrita na seção 2.3, para obter um algoritmo polinomial para calcular $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ de qualquer $(q, q - 4)$ -grafo G , quando q é um inteiro fixo. Para tanto iremos utilizar uma abordagem recursiva em que o número lid-cromático e sua versão forte são ambos calculados para cada nó interno (que representa um subgrafo de G) da árvore de decomposição, baseados nos resultados obtidos para os seus dois filhos. As folhas da decomposição serão tratadas no teorema final desta seção. Repetimos este procedimento até calcular χ_{lid} e $\widetilde{\chi}_{lid}$ para o grafo representado pela raiz da árvore, que é o próprio G .

A abordagem descrita acima foi utilizada, por exemplo, para calcular o número cromático acíclico, o número cromático estrela, o número cromático não repetitivo e o número cromático de clique de $(q, q-4)$ -grafos, bem como o número harmônico de $(q, q-4)$ -grafos conexos (LINHARES-SALES *et al.*, 2014).

Iniciamos mostrando como calcular o valor de $\widetilde{\chi}_{lid}(G)$ quando G é a união de dois grafos.

Lema 3.2 *Seja um grafo $G = G_1 \cup G_2$, então:*

$$\widetilde{\chi}_{lid}(G) = \begin{cases} \max\{\widetilde{\chi}_{lid}(G_1), \widetilde{\chi}_{lid}(G_2)\}, & \text{se } G_1 \text{ e } G_2 \text{ não possuem vértices universais;} \\ \max\{\widetilde{\chi}_{lid}(G_1) + 1, \widetilde{\chi}_{lid}(G_2)\}, & \text{se apenas } G_1 \text{ possui vértices universais;} \\ \max\{\widetilde{\chi}_{lid}(G_1), \widetilde{\chi}_{lid}(G_2) + 1\}, & \text{se apenas } G_2 \text{ possui vértices universais;} \\ \max\{\widetilde{\chi}_{lid}(G_1) + 1, \widetilde{\chi}_{lid}(G_2) + 1\}, & \text{se } G_1 \text{ e } G_2 \text{ possuem vértices universais.} \end{cases}$$

Prova: Suponha que G_1 e G_2 não possuem vértices universais e, sem perda de generalidade (s.p.d.g.), que $\widetilde{\chi}_{lid}(G_1) \geq \widetilde{\chi}_{lid}(G_2)$. Certamente não podemos ter uma coloração com menos que $\widetilde{\chi}_{lid}(G_1)$ cores, além disso para obter uma coloração do grafo G basta colorirmos G_1 e G_2 separadamente uma vez que não há nenhum vértice ruim nas colorações de G_1 e G_2 e não existem arestas entre G_1 e G_2 .

Agora suponha que apenas G_1 possui vértices universais. Se $\widetilde{\chi}_{lid}(G_2) \geq \widetilde{\chi}_{lid}(G_1) + 1$, então basta colorirmos G_1 e G_2 separadamente como fizemos no caso anterior, pois ao menos uma cor não estará na vizinhança de nenhum vértice de G_1 (i.e. a cor $\widetilde{\chi}_{lid}(G_1) + 1$). Além disso, como G_2 não possui vértice universal, então nenhum

vértice terá todas as cores em sua vizinhança. Se $\widetilde{\chi}_{lid}(G_2) < \widetilde{\chi}_{lid}(G_1) + 1$, então em uma slid-coloração de G com $\widetilde{\chi}_{lid}(G_1)$ cores, todas as cores devem aparecer em vértices de G_1 e, como G_1 possui um vértice universal, tal vértice possui todas as cores em sua vizinhança e não é universal, pois não é adjacente a nenhum vértice de G_2 e, portanto, a coloração não seria forte. Para obter uma coloração de G com $\widetilde{\chi}_{lid}(G_1) + 1$ cores basta colorir cada cada grafo separadamente mudando uma das classes de cor em G_1 ou em G_2 para uma cor ainda não utilizada na coloração (i.e. a cor $\widetilde{\chi}_{lid}(G_1) + 1$). O caso em que apenas G_2 possui vértices universais é análogo ao tratado no parágrafo anterior.

Caso G_1 e G_2 possuam vértices universais, suponha s.p.d.g. que $\widetilde{\chi}_{lid}(G_1) \geq \widetilde{\chi}_{lid}(G_2)$. Novamente não podemos ter uma slid-coloração de G com $\widetilde{\chi}_{lid}(G_1)$ cores, uma vez que, desta maneira o vértice universal em G_1 teria todas as cores em sua vizinhança, e não seria universal em G . Assim, utilizando o mesmo procedimento anterior, colorimos cada grafo separadamente e alteramos uma das classes de cor em um deles para uma cor não utilizada. Desta maneira obtemos uma slid-coloração de G com $\widetilde{\chi}_{lid}(G_1) + 1$ cores. \square

Lema 3.3 *Seja um grafo $G = G_1 \vee G_2$, então $\widetilde{\chi}_{lid}(G) = \widetilde{\chi}_{lid}(G_1) + \widetilde{\chi}_{lid}(G_2)$.*

Prova: Primeiramente, observe que qualquer slid-coloração de G deve induzir uma slid-coloração em G_1 e G_2 . Além disso as cores utilizadas para colorir os vértices de G_1 devem ser diferentes das cores utilizadas para colorir os vértices de G_2 . Portanto, $\widetilde{\chi}_{lid}(G) \geq \widetilde{\chi}_{lid}(G_1) + \widetilde{\chi}_{lid}(G_2)$.

Resta-nos mostrar uma slid-coloração de G com $\widetilde{\chi}_{lid}(G_1) + \widetilde{\chi}_{lid}(G_2)$ cores. Dada uma slid-coloração mínima c_1 de G_1 e uma slid-coloração mínima c_2 de G_2 , definimos uma slid-coloração c de G como abaixo.

$$c(v) = \begin{cases} c_1(v), & \text{se } v \in V(G_1); \\ c_2(v) + \widetilde{\chi}_{lid}(G_1), & \text{se } v \in V(G_2). \end{cases}$$

A coloração c certamente é própria, pois ela é própria em G_1 e G_2 separadamente e os vértices de G_1 são coloridos com cores distintas dos vértices de G_2 .

Observe que se dois vértices adjacentes de G_1 tem vizinhanças fechadas distintas em G , então eles possuem vizinhanças fechadas distintas em G_1 e, portanto, terão cores distintas em suas vizinhanças fechadas na coloração c_1 e também em c . Esta mesma argumentação pode ser aplicada de maneira análoga a G_2 e c_2 . Observe ainda que dados $v_1 \in V(G_1)$ e $v_2 \in V(G_2)$, temos que v_1 (v_2) tem todas as cores utilizadas em G_2 (G_1) em sua vizinhança fechada, portanto $c(N[v_1]) = c(N[v_2])$ se, e somente se, ambos os vértices são universais, logo c é uma lid-coloração.

Da mesma maneira, um vértice de G_1 (G_2) é universal em G se, e somente se, ele é universal em G_1 (G_2), portanto, qualquer vértice que não seja universal em G não possui todas as cores de c em sua vizinhança fechada. Logo c é uma slid-coloração. \square

Lema 3.4 *Seja um grafo $G = G_1 \cup G_2$, então $\chi_{lid}(G) = \max\{\chi_{lid}(G_1), \chi_{lid}(G_2)\}$.*

Prova: Note que qualquer lid-coloração de G deve induzir uma lid-coloração em G_1 e uma lid-coloração em G_2 . Ademais as cores utilizadas em vértices de G_1 podem ser utilizadas em vértices de G_2 e vice-versa. Portanto, $\chi_{lid}(G) \geq \max\{\chi_{lid}(G_1), \chi_{lid}(G_2)\}$.

Além disso dada uma lid-coloração c_1 de G_1 com cores $\{1, \dots, k_1\}$ e uma lid-coloração c_2 de G_2 com cores $\{1, \dots, k_2\}$. A coloração c de G que atribui a cada vértice v_1 de G_1 a cor $c_1(v_1)$ e a cada vértice v_2 de G_2 a cor $c_2(v_2)$ é uma lid-coloração de G , uma vez que toda aresta de G possui ambas as extremidades em G_1 ou ambas as extremidades em G_2 . Portanto, $\chi_{lid}(G) \leq \max\{\chi_{lid}(G_1), \chi_{lid}(G_2)\}$. \square

No lema abaixo iremos calcular o valor de $\chi_{lid}(G)$ quando G é a junção de dois grafos. Observe, que neste caso o $\chi_{lid}(G)$ depende de $\widetilde{\chi}_{lid}(G_1)$ e $\widetilde{\chi}_{lid}(G_2)$.

Lema 3.5 *Sejam G_1 e G_2 dois grafos e $G = G_1 \vee G_2$. Se G_1 e G_2 não possuem nenhum vértice universal, então $\chi_{lid}(G) = \min\{\widetilde{\chi}_{lid}(G_1) + \chi_{lid}(G_2), \widetilde{\chi}_{lid}(G_2) + \chi_{lid}(G_1)\}$. Caso contrário, $\chi_{lid}(G) = \widetilde{\chi}_{lid}(G_1) + \widetilde{\chi}_{lid}(G_2)$.*

Prova:

Se G_1 ou G_2 possuírem um vértice universal u , então u também será universal em G . Como já mencionamos na Seção 3.2, para todo grafo G' que possui um vértice universal, temos que $\chi_{lid}(G') = \widetilde{\chi}_{lid}(G')$. Assim, se G_1 ou G_2 possuírem um vértice universal, pelo Lema 3.3, $\chi_{lid}(G) = \widetilde{\chi}_{lid}(G) = \widetilde{\chi}_{lid}(G_1) + \widetilde{\chi}_{lid}(G_2)$.

Agora assumamos que G_1 e G_2 não possuem vértices universais. Observe que em uma lid-coloração de G , vértices em G_1 devem receber cores distintas de vértices de G_2 . Além disso qualquer lid-coloração de G deve induzir lid-colorações em G_1 e G_2 , visto que dois vértices com vizinhanças distintas em G_1 ou G_2 também possuem vizinhanças distintas em G .

Sejam c_1 e c_2 lid-colorações de G_1 e G_2 , respectivamente. Seja ainda c a coloração de G que atribui a cada vértice v_1 de G_1 a cor $c_1(v_1)$ e a cada vértice v_2 de G_2 a cor $c_2(v_2)$. Se nem c_1 nem c_2 forem slid-colorações, então existe um vértice v_1 de G_1 em contato com todas as cores de c_1 e existe um vértice v_2 de G_2 em contato com todas as cores de c_2 . Desta maneira $c(N[v_1]) = c(N[v_2])$ no entanto, como nem v_1 nem v_2 são vértices universais temos que $N[v_1] \neq N[v_2]$, o que viola a condição (ii). Portanto, se G_1 e G_2 não possuem vértices universais, uma lid-coloração de G deve induzir em pelo menos um desses dois grafos uma slid-coloração. Assim temos que $\chi_{lid}(G) \geq \min\{\widetilde{\chi}_{lid}(G_1) + \chi_{lid}(G_2), \widetilde{\chi}_{lid}(G_2) + \chi_{lid}(G_1)\}$.

Resta-nos mostrar uma lid-coloração de G com $\min\{\widetilde{\chi}_{lid}(G_1) + \chi_{lid}(G_2), \widetilde{\chi}_{lid}(G_2) + \chi_{lid}(G_1)\}$ cores.

Suponha novamente que G_1 e G_2 não possuem vértices universais e sejam c_1 uma lid-coloração mínima de G_1 com cores $\{1, \dots, \chi_{lid}(G_1)\}$ e c_2 uma slid-coloração de G_2 com cores $\{\chi_{lid}(G_1) + 1, \dots, \chi_{lid}(G_1) + \widetilde{\chi}_{lid}(G_2)\}$. Assim como fizemos acima, seja c a coloração de G que atribui a cada vértice v_1 de G_1 a cor $c_1(v_1)$ e a cada vértice v_2 de

G_2 a cor $c_2(v_2)$. Certamente c é uma coloração própria de G , pois c_1 e c_2 são próprias em G_1 e G_2 , respectivamente, e as cores utilizadas em vértices de G_1 são distintas das cores utilizadas em vértices de G_2 . Como G_2 não possui vértices universais e c_2 é uma slid-coloração, então nenhum vértice de G_2 irá ter contato com todas as cores de c_2 e portanto não poderá ter o mesmo conjunto de cores que nenhum vértice de G_1 , uma vez que para todo vértice v_1 de G_1 temos que $\{\chi_{lid}(G_1)+1, \dots, \chi_{lid}(G_1)+\widetilde{\chi}_{lid}(G_2)\} \subset c(N[v_1])$. Ademais se tomarmos dois vértices u e v adjacentes em G_1 (G_2) se suas vizinhanças fechadas forem distintas em G também o serão em G_1 (G_2), como c_1 (c_2) é uma (s)lid-coloração de G_1 (G_2), então as cores das vizinhanças fechadas dos vértices serão distintas também em c .

O raciocínio é análogo se considerarmos uma slid-coloração de G_1 e uma lid-coloração de G_2 . Assim, se G_1 e G_2 não possuem vértices universais, então $\chi_{lid}(G) = \min\{\widetilde{\chi}_{lid}(G_1) + \chi_{lid}(G_2), \widetilde{\chi}_{lid}(G_2) + \chi_{lid}(G_1)\}$.

□

O lema abaixo nos mostra como calcular $\chi_{lid}(G)$ e $\widetilde{\chi}_{lid}(G)$ quando G é uma aranha magra.

Lema 3.6 *Seja G um grafo aranha magra com partição (R, C, S) . Se G não for um P_4 , então:*

$$\chi_{lid}(G) = \begin{cases} \widetilde{\chi}_{lid}(G[R]) + |C| + |S| - 1, & \text{se } G[R] \text{ não possui um vértice universal;} \\ \chi_{lid}(G[R]) + |C| + |S|, & \text{Caso contrário.} \end{cases}$$

Além disso, se $|C| = 2$, então $\widetilde{\chi}_{lid}(G) = \chi_{lid}(G[R]) + |C| + |S|$. Caso contrário, $\widetilde{\chi}_{lid}(G) = \chi_{lid}(G)$.

Prova: Uma lid-coloração de G deve induzir uma lid-coloração de $G[R]$ uma vez que dois vértices de R têm vizinhanças distintas em $G[R]$ se, e somente se, também tem vizinhanças distintas em G . Tal coloração deve atribuir cores distintas para os vértices de C e de R , caso contrário não seria própria. Pelo mesmo motivo, todos os vértices de C devem receber cores distintas.

Argumentamos que os vértices de S também devem ter cores distintas entre si em uma lid-coloração c de G . Basta ver que se dois vértices $s_i, s_j \in S$ recebem a mesma cor em uma lid-coloração de G , então teríamos dois vértices $c_i, c_j \in C$ adjacentes tais que $N[c_i] \neq N[c_j]$ e $c(N[c_i]) = c(N[c_j])$, uma violação à condição (ii). Pelo mesmo motivo, no máximo um vértice de S pode receber uma cor já utilizada em vértices de C ou de R .

Ademais, suponha que existe um vértice universal u em $G[R]$ e em uma lid-coloração de G um vértice $s_i \in S$ recebe uma cor já utilizada em um vértice de C ou de R . Assim teríamos dois vértices adjacentes u e c_i tais que $N[u] \neq N[c_i]$ mas $c(N[c_i]) = c(N[s_i]) = c(R) \cup c(C)$, novamente uma violação da condição (ii). De fato, para podermos reutilizar uma cor de $c(C)$ ou $c(R)$ em um vértice de S temos que $G[R]$

não pode ter um vértice universal e c deve induzir uma slid-coloração de $G[R]$. Sendo assim:

$$\chi_{lid}(G) \geq \begin{cases} \widetilde{\chi}_{lid}(G[R]) + |C| + |S| - 1, & \text{se } G[R] \text{ não possui um vértice universal;} \\ \chi_{lid}(G[R]) + |C| + |S|, & \text{Caso contrário.} \end{cases}$$

Para construir uma lid-coloração de uma aranha magra com esta quantidade de cores basta colorir os vértices de R com um (s)lid-coloração ótima de $G[R]$ e atribuir cores distintas das utilizadas aos vértices de C e S , com a possível exceção de um vértice de S receber uma cor já utilizada em $C \cup R$ (caso $G[R]$ não tenha vértice universal).

Finalmente, observe ainda que, quando $|C| \neq 2$, nenhum vértice possui todas as cores em sua vizinhança fechada na lid-coloração acima, ou seja esta é uma slid-coloração. Quando $|C| = 2$, caso reutilizemos uma cor de um vértice em $C \cup R$ em um vértice $s_i \in S$, então o vértice de C não-adjacente a s_i seria um vértice ruim. Para que isso não ocorra basta recolorir s_i com uma cor distinta de todos os demais vértices. Portanto, quando $|C| = 2$, $\widetilde{\chi}_{lid}(G) = \chi_{lid}(G[R]) + |C| + |S|$ \square

Como uma aranha gorda com $|C| = 1$ é desconexa e com $|C| = 2$ é também uma aranha magra, podemos assumir que $|C| \geq 3$ no Lema abaixo.

Lema 3.7 *Seja G um grafo aranha gorda com partição (R, C, S) , então $\chi_{lid}(G) = \chi_{lid}(G[R]) + |C| + |S| - 1$ e $\widetilde{\chi}_{lid}(G) = \chi_{lid}(G[R]) + |C| + |S|$.*

Prova: Uma lid-coloração de G deve induzir uma lid-coloração de $G[R]$ uma vez que dois vértices de R têm vizinhanças distintas em $G[R]$ se, e somente se, também tem vizinhanças distintas em G . Tal coloração deve atribuir cores distintas para os vértices de C e de R , caso contrário não seria própria. Pelo mesmo motivo, todos os vértices de C devem receber cores distintas.

Os vértices de S também devem ter cores distintas entre si em uma lid-coloração c de G , senão vejamos. Suponha que dois vértices $s_i, s_j \in S$ recebem a mesma cor em uma lid-coloração de G , assim teríamos dois vértices $c_i, c_j \in C$ adjacentes tais que $N[c_i] \neq N[c_j]$ e $c(N[c_i]) = c(N[c_j])$ uma violação à condição (ii) e, portanto, uma contradição. Pelo mesmo motivo, no máximo um vértice de S pode receber uma cor já utilizada em vértices de C ou de R .

Uma lid-coloração de G é obtida colorindo os vértices de R de maneira ótima em $G[R]$, os vértices de C com $|C|$ cores distintas das demais e os vértices de S com $|S| - 1$ cores distintas das demais, mais uma cor já utilizada em vértices de C ou R . Tal coloração não é slid-coloração de G , uma vez que o vértice de C não adjacente ao vértice de S cuja cor foi reaproveitada é um vértice ruim. Para a coloração se tornar slid basta recolorirmos tal vértice de S com uma nova cor. \square

No lema abaixo iremos utilizar a notação $|c|$ como o número de cores de uma

coloração c . Este lema nos mostra como construir uma lid-coloração e uma slid-coloração de um grafo G quando este possui uma p -componente separável com menos que q vértices.

Lema 3.8 *Seja q um inteiro fixo e $G = (V, E)$ um grafo que contém uma p -componente separável $H = (H_1, H_2)$ com menos que q vértices tal que todo vértice de $G - H$ é adjacente aos vértices de H_1 e não-adjacente aos vértices de H_2 . Dados $\chi_{lid}(G - H)$ e $\widetilde{\chi}_{lid}(G - H)$ (com suas respectivas colorações ótimas) podemos calcular uma lid-coloração e uma lid-coloração forte, ambas ótimas, de G em tempo $O(q^{2q+5})$.*

Prova: Observe que toda lid-coloração de G induz uma lid-coloração de $G - H$, uma vez que todos os vértices de $G - H$ possuem o mesmo conjunto de vizinhos em H . Assim qualquer lid-coloração de G utiliza pelo menos $\chi_{lid}(G - H)$ cores nos vértices de $G - H$.

Primeiramente suponha que $\chi_{lid}(G - H) \geq q$. Construimos o grafo G' a partir de H , incluindo uma clique $\{x_1, \dots, x_q\}$ e ligando todos os vértices dessa clique com os vértices de H_1 . Como G' possui menos que $2q$ vértices, podemos gerar todas as colorações possíveis de G' com as cores $\{1, \dots, 2q\}$ e verificar se são uma lid-coloração. Isto pode ser feito em tempo $O(q^{2q+4})$, iniciamos colorindo os vértices x_1, \dots, x_q com as cores $1, \dots, q$, respectivamente, e, para todo vértice de H , temos no máximo $2q$ cores disponíveis, resultando em $(2q)^q$ colorações. Finalmente, para cada coloração possível, devemos verificar se a mesma é, de fato, uma lid-coloração, o que pode ser feito em tempo $O((2q)^4)$. Como $q \geq 4$, então $(2q)^{q+4} \leq (q^2/2)^q \cdot (2q)^4 \leq q^{2q+4}$. Como o processo é exaustivo, podemos obter uma lid-coloração mínima de G' nesse tempo.

Para simplificar a notação, assumiremos sem perda de generalidade que as colorações utilizadas satisfazem as seguintes restrições. Dadas colorações c, c', c'' de $G, G', G - H$, respectivamente:

$$(R1) \quad c(G - H) = \{1, \dots, |c(G - H)|\} \text{ e } c(G - H) \cap c(H) = \{1, \dots, |c(G - H) \cap c(H)|\};$$

$$(R2) \quad c'' = \{1, \dots, |c''|\};$$

$$(R3) \quad c' = \{1, \dots, q\} \cup \{|c''| + 1, \dots, |c''| + |c'| - q\} \text{ e } c'(x_i) = i \text{ para todo } i \in \{1, \dots, q\}.$$

Dadas colorações c', c'' de $G', G - H$, respectivamente, satisfazendo (R2) e (R3), dizemos que (c', c'') induz a coloração c de G definida a seguir, que satisfaz (R1). Escrevemos $(c', c'') \rightarrow c$. Se $v \notin H$, então $c(v) = c''(v)$; se $v \in H$, então $c(v) = c'(v)$.

Por outro lado, seja c uma coloração de G que satisfaz (R1). Note que $|c(G - H) \cap c(H)| < q$, pois H tem menos de q vértices. Dizemos que c induz as colorações c', c'' de G' e $G - H$, respectivamente, que satisfazem (R2) e (R3) descritas a seguir. Escrevemos $c \rightarrow (c', c'')$. Seja $c''(v) = c(v)$ para todo vértice v de $G - H$. Seja $c'(x_i) = i$ para todo $1 \leq i \leq q$ e, para todo $v \in H$, $c'(v) = c(v)$.

Note que $c \rightarrow (c', c'')$ se e somente $(c', c'') \rightarrow c$. Por isso, podemos escrever $(c', c'') \leftrightarrow c$.

Observe que, se c é uma lid-coloração, então c'' é uma lid-coloração e c' é uma coloração própria que pode não ser uma lid-coloração mas diferencia os vértices em H (ou seja, $c'(N[v_1]) \neq c'(N[v_2])$ para toda aresta v_1v_2 de H tal que $N[v_1] \neq N[v_2]$). Assim, a

questão é saber se c' diferencia vértices entre H_1 e $\{x_1, \dots, x_q\}$ (lembre que todo vértice de H_1 tem um vizinho em H_2).

Queremos mostrar que, se c'' é uma lid-coloração de $G - H$ e c' é uma coloração própria de G' com certas propriedades, então c , tal que $(c', c'') \rightarrow c$, é uma lid-coloração de G . Por outro lado, se c é uma lid-coloração de G , então c'' é uma lid-coloração de $G - H$ e c' é uma coloração própria de G' com certas propriedades.

Seja c uma lid-coloração de G que satisfaz (R1) e sejam c' e c'' tais que $c \rightarrow (c', c'')$. Como $\chi_{lid}(G - H) \geq q$, então $|c''| \geq q$.

Suponha que $G - H$ tem um vértice u que é universal em $G - H$ ou c'' não é uma lid-coloração forte (ou seja, existe um vértice não-universal u de $G - H$ tal que $c''(N[u]) = c''$). Então c' deve ser uma lid-coloração de G' , pois, para todo vértice v de H_1 , $c(N[v]) \neq c(N[u]) = c'' \cup c(H_1)$ e portanto $c'(N[v]) \neq \{1, \dots, q\} \cup c'(H_1) = c'(x_1)$ (lembre que todo vértice de H_1 tem um vizinho em H_2). Suponha agora que $G - H$ não tem vértice universal e c'' é uma lid-coloração forte. Portanto, como $c'' \subseteq c(N[v])$ para todo $v \in H_1$, então c' não necessariamente diferencia vértices entre H_1 e $\{x_1, \dots, x_q\}$.

Por outro lado, sejam c', c'' colorações próprias de $G', G - H$ satisfazendo (R2) e (R3) tais que c'' é uma lid-coloração e c' diferencia os vértices em H (ou seja, $c'(N[v_1]) \neq c'(N[v_2])$ para toda aresta v_1v_2 de H tal que $N[v_1] \neq N[v_2]$).

Seja c a coloração de G tal que $(c', c'') \rightarrow c$. Suponha que $G - H$ tem um vértice u que é universal em $G - H$ ou c'' não é uma lid-coloração forte. Portanto, se c' é uma lid-coloração de G' , então c é uma lid-coloração de G , pois, para todo vértice v de H_1 , $c'' \subseteq c(N[v])$ e $c'(N[v]) \neq \{1, \dots, q\} \cup c'(H_1) = c'(x_1)$ e portanto $c(N[v]) \neq c'' \cup c(H_1) = c(N[u])$. Suponha agora que $G - H$ não tem vértice universal e c'' é uma lid-coloração forte. Portanto, como c' diferencia os vértices em H e $c(N[u]) \subsetneq c'' \subseteq c(N[v])$ para todo $v \in H_1$ e $u \in V(G - H)$, então c é uma lid-coloração de G .

Resumindo, sejam c, c', c'' colorações de G, G' e $G - H$, respectivamente, satisfazendo (R1), (R2) e (R3) tais que $(c', c'') \leftrightarrow c$. Então, c é uma lid-coloração de G se e somente se c'' é uma lid-coloração de $G - H$ e:

- (a) c' é uma coloração própria de G' que diferencia apenas os vértices em H , se $G - H$ não tem vértice universal e c'' é uma lid-coloração forte; ou
- (b) c' é uma lid-coloração de G' , caso contrário.

Com isso, dada uma lid-coloração c'' de $G - H$ satisfazendo (R2), podemos obter o seguinte algoritmo para gerar uma lid-coloração c de G tal que c induz c'' com número mínimo de cores. Se c'' é uma lid-coloração forte e $G - H$ não tem vértice universal, então, para cada coloração própria c'_i de G' satisfazendo (R3) e que diferencia os vértices em H , obtenha a coloração c_i de G tal que $(c'_i, c'') \rightarrow c_i$. Caso contrário, para cada lid-coloração c'_i de G' , obtenha a coloração c_i de G tal que $(c'_i, c'') \rightarrow c_i$. Retorne a lid-coloração c_i com número mínimo de cores. Esse mesmo procedimento pode retornar uma lid-coloração forte de G (basta rejeitar as lid-colorações c_i geradas que não são fortes).

Portanto, se $\widetilde{\chi}_{lid}(G - H) = \chi_{lid}(G - H)$, faça c'' uma lid-coloração forte ótima de $G - H$. Se $\widetilde{\chi}_{lid}(G - H) > \chi_{lid}(G - H)$, faça c'_1 uma lid-coloração ótima de $G - H$ e c'_2 uma lid-coloração forte ótima de $G - H$ e aplique o algoritmo para c'_1 e c'_2 retornando a coloração c de G com menor número de cores. Lembre que, pelo Teorema 3.1, sempre podemos obter uma lid-coloração forte a partir de uma lid-coloração não-forte com no máximo uma cor adicional.

Se $\chi_{lid}(G - H) < q$, o argumento é semelhante: ao invés de termos um grafo G' , teremos uma família de grafos $G'_{\chi_{lid}(G-H)}, \dots, G'_q$, onde cada G'_p é obtido da mesma maneira que G' substituindo a clique de tamanho q por uma de tamanho p . Da mesma forma, é possível calcular as lid-colorações para cada um desses grafos, o que pode ser feito em tempo $O(q^{2q+5})$. Devemos fazer desta maneira pois uma lid-coloração mínima c de G não necessariamente induz uma lid-coloração mínima c'' de $G - H$ \square

Teorema 3.4 *Dado um $(q, q-4)$ -grafo G com n vértices e m arestas podemos determinar $\widetilde{\chi}_{lid}(G)$ e $\chi_{lid}(G)$, bem como colorações ótimas para ambos os casos, em tempo $O(q^{2q+5}n + m)$.*

Prova: Primeiro calculamos a árvore de decomposição primeval de G , T_G , em tempo $O(m + n)$. Cada folha G' de T_G é um subgrafo de G com menos do que q vértices ou uma aranha sem cabeça. No caso de G' ser um grafo aranha sem cabeça, podemos calcular $\chi_{lid}(G')$ e $\widetilde{\chi}_{lid}(G')$ em tempo constante utilizando um dos Lemas 3.6 e 3.7, dependendo se G' é uma aranha magra ou gorda. Se G' for um grafo com menos que q vértices podemos gerar todas as colorações com no máximo q cores, e verificar se elas são (s)lid-coloração em tempo $O(q^{q+3})$.

Cada vértice interno G' de T_G é um subgrafo de G resultante de operações (\vee , \cup ou \uplus) sobre os seus filhos. Assim, se G' é o resultado de uma operação \cup podemos calcular $\chi_{lid}(G')$ e $\widetilde{\chi}_{lid}(G')$ em tempo constante utilizando os Lemas 3.2 e 3.4. Se G' é o resultado de uma operação \vee podemos calcular $\chi_{lid}(G')$ e $\widetilde{\chi}_{lid}(G')$ em tempo constante utilizando os Lemas 3.3 e 3.5. Se G' é o resultado de uma operação \uplus podemos calcular $\chi_{lid}(G')$ e $\widetilde{\chi}_{lid}(G')$ em tempo $O(q^{2q+5})$ utilizando os Lemas 3.6, 3.7 e 3.8.

Como T_G possui $O(n)$ vértices, todo o processo, até chegar na raiz G de T_G , leva tempo $O(q^{2q+5}n + m)$. \square

A Teorema 3.4 ainda implica um algoritmo FPT para os problemas de lid-coloração e slid-coloração, quando consideramos q um parâmetro fixo.

3.5 Lid-Coloração em grafos com largura em árvore limitada

Como mencionamos na Seção 2.4, Bodlaender descreveu uma técnica para resolução de problemas em grafos com largura em árvore limitada (BODLAENDER, 1997) utilizando sua decomposição em árvore. Nesta técnica, dado uma decomposição em árvore de um grafo G , T_G com raiz r , para cada nó v , iniciando pelas folhas, calculamos uma

tabela com soluções parciais do problema referente ao subgrafo induzido pelos vértices de G que pertencem aos nós da subárvore de T_G com raiz v . Cada tabela é calculada utilizando somente as tabelas dos filhos de v e a informação de G restrita aos vértices pertencentes a v . A solução do problema em relação ao grafo G deve ser encontrada na tabela de r .

Utilizaremos a técnica descrita por Bodlaender para determinar, dado um grafo G e um inteiro k , se $\chi_{lid}(G) \leq k$ e $\widetilde{\chi}_{lid}(G) \leq k$.

Seja uma decomposição em árvore boa $D = (\mathcal{X} = X_i | i \in I, T = (I, F))$ com raiz r de um grafo G com largura em árvore w . Para $t \in I$ definimos G_t como o grafo induzido em G pelos vértices pertencentes aos nós da subárvore de T com raiz t .

Para cada nó $t \in I$ computamos uma tabela W_t com entradas (ψ, f, g) , onde ψ é uma coloração própria dos vértices de X_t , $f : X_t \rightarrow \mathcal{P}([k])$ é uma função que, para cada vértice $v \in X_t$, indica as cores na vizinhança de v , para uma coloração de G_t que estende ψ , e $g : X_t \rightarrow \mathcal{P}(\mathcal{P}([k]))$ é uma função que, para cada vértice $v \in X_t$, indica os conjuntos de cores, para esta mesma coloração de G_t que estende ψ , na vizinhança fechada de todo vizinho $w \in V(G_t) \setminus X_t$ tal que $N[v] \neq N[w]$. Ou seja, $g(v)$ é o conjunto dos conjuntos de cores proibidos para v por seus vizinhos já esquecidos na subárvore com raiz t na coloração de G_t que estende ψ .

Como $|X_t| \leq w$, então o número de colorações ψ é limitado por k^w , uma vez que cada vértice de X_t pode receber uma dentre as k cores possíveis. Para limitar a quantidade de funções f possíveis, basta observar que para cada vértice de X_t associamos um elemento de $\mathcal{P}([k])$, como $|\mathcal{P}([k])| = 2^k$ temos que o número máximo de funções f distintas em X_t é limitado por $(2^k)^w$. Por fim, para limitar a quantidade de funções g basta observar que para cada vértice de X_t associamos um elemento de $\mathcal{P}(\mathcal{P}([k]))$ como $|\mathcal{P}(\mathcal{P}([k]))| = 2^{2^k}$ temos que o número máximo de funções f distintas em X_t é limitado por $(2^{2^k})^w$. Assim, podemos concluir que cada tabela W_t possui tamanho no máximo $(k2^{k+2^k})^w$.

Uma entrada (ψ, f, g) de um nó $t \in T$ é considerada uma entrada *boa*, se para todo $v \in X_t$, $f(v) \not\subseteq g(v)$ (se v não está em contato com o mesmo conjunto de cores de algum vizinho $w \in V(G_t) \setminus X_t$ tal que $N[v] \neq N[w]$) e para todo par de vértices adjacentes $w, v \in X_t$ se $N[v] \neq N[w]$ então $f(v) \neq f(w)$ (se v não está em contato com o mesmo conjunto de cores de algum vizinho $w \in V(G_t)$). Caso contrário (ψ, f, g) é dita uma entrada *ruim*.

Por construção, o grafo G_t possui uma k -lid-coloração se o nó t possui ao menos uma entrada boa dentre todas as entradas de W_t . Em particular, G possui um lid-coloração com k cores, se W_r possui ao menos uma entrada boa.

Nosso procedimento se resume a calcular todas as possíveis entradas nos nós folha e, a partir destes, calcular as entradas dos nós acima até chegarmos na raiz de T onde poderemos determinar checando cada uma das entradas se há uma entrada boa e,

portanto, uma lid-coloração do grafo em questão com k cores. Agora vamos determinar como calcular W_t dependendo do tipo do nó t .

Lema 3.9 *Se t é um nó folha, então podemos calcular W_t em tempo $O(k)$.*

Prova: Seja $X_t = \{v\}$. Iremos criar uma entrada para cada cor $i \in \{1, \dots, k\}$ onde, $\psi(v) = i$, $f(v) = \{i\}$ e $g(v) = \emptyset$, uma vez que v não possui vizinhos que proíbam certos subconjuntos de cores em G_t . \square

A seguir iremos tratar dos nós do tipo saída.

Lema 3.10 *Seja t um nó do tipo saída com filho t' , tal que $X_t = X_{t'} \setminus \{w\}$. Podemos calcular W_t em tempo $O\left(\left(k2^{k+2^k}\right)^w\right)$.*

Prova: Para cada entrada (ψ', f', g') de $W_{t'}$ iremos construir a entrada (ψ, f, g) de W_t , tal que as funções ψ e f são, respectivamente, restrições de ψ' e f' ao conjunto X_t . Além disso a função g é definida da seguinte maneira:

$$g(v) = \begin{cases} g'(v), & \text{se } v \notin N(w); \\ g'(v) \cup \{(f'(w))\}, & \text{Caso contrário.} \end{cases}$$

A análise de tempo decorre do limite do tamanho da tabela de $X_{t'}$. \square

Lema 3.11 *Seja t um nó do tipo introdução com filho t' , tal que $X_t = X_{t'} \cup \{w\}$. Podemos calcular W_t em tempo $O\left(k^{w+1} \left(2^{k+2^k}\right)^w\right)$.*

Prova: Para cada entrada (ψ', f', g') de $W_{t'}$ e para cada cor $i \in 1, \dots, k$ o que vamos fazer é expandir a coloração ψ' para o conjunto X_t colorindo w com a cor i e deixando os demais vértices com suas cores em ψ' . Se a coloração gerada ψ não for uma coloração própria de $G[X_t]$ descartamos esta combinação entrada/cor e seguimos o processo. Caso ψ seja uma coloração própria definimos a função f da seguinte maneira:

$$f(v) = \begin{cases} f'(v), & \text{se } v \notin N(w); \\ \psi(N(w)) \cup \{i\}, & \text{se } v = w; \\ f'(v) \cup \{i\}, & \text{Caso contrário.} \end{cases}$$

A função $g(v)$ é igual a $g'(v)$ se $v \neq w$ e é vazia para w . A análise de tempo decorre do limite do tamanho da tabela de $X_{t'}$ e do fato que para cada entrada da mesma calculamos k variantes. \square

Lema 3.12 *Seja t um nó do tipo junção com filhos t_1 e t_2 podemos calcular W_t em tempo $O\left(\left(k2^{k+2^k}\right)^{2w}\right)$.*

Prova: Para cada par de entradas $E_1 = (\psi_1, f_1, g_1)$ de W_{t_1} e $E_2 = (\psi_2, f_2, g_2)$ de W_{t_2} . Caso $\psi_1(v) \neq \psi_2(v)$ para algum $v \in X_t$, isto é ψ_1 e ψ_2 representam colorações distintas dos vértices de X_t , então descartamos esta combinação de entradas e continuamos o processo. Caso contrário, definimos a entrada (ψ, f, g) em W_t : para todo $v \in X_t$ $\psi(v) = \psi_1(v) =$

$\psi_2(v)$, $f(v) = f_1(v) \cup f_2(v)$ e $g(v) = g_1(v) \cup g_2(v)$.

A análise do tempo decorre do limite do tamanho de W_{t_1} e W_{t_2} e do fato que realizamos os cálculos acima para cada par de entradas. \square

Teorema 3.5 *Dado um grafo G com n vértices e um inteiro k , pode-se determinar se $\chi_{lid}(G) \leq k$ e $\widetilde{\chi}_{lid}(G) \leq k$ em tempo $O\left(n \left(k2^{k+2^k}\right)^{2w}\right)$.*

Prova: No caso de $\chi_{lid}(G)$ o resultado segue diretamente dos Lemas 3.9, 3.10, 3.11, 3.12 e do Teorema 2.4, uma vez que uma decomposição em árvore boa de um grafo com n vértices possui $O(n)$ nós.

Além disso, observe que poderíamos modificar o processo para descartamos uma entrada, sempre que esta tiver um vértice v não universal em G tal que $f(v) = [k]$, para calcular $\widetilde{\chi}_{lid}(G)$. \square

Gonçalves et al. mencionam que se G possui largura em árvore limitada, então $\chi_{lid}(G)$ é limitado (GONÇALVES, PARREAU, and PINLOU, 2013). Observe que se pudermos encontrar uma função que limita $\chi_{lid}(G)$ por meio da largura em árvore de G o resultado acima implica em um algoritmo FPT para calcular o número lid-cromático de grafos quando consideramos a largura em árvore com parâmetro fixo.

4 JOGO DE POLÍCIA E LADRÃO EM GRAFOS

Neste capítulo iremos estudar o jogo de polícia e ladrão em grafos, especificamente em dois parâmetros deste jogo: o *cop-number* e o *capture time*. Como ambos os parâmetros são difíceis de determinar para grafos em geral, iremos utilizar as decomposições apresentadas na Seção 2.3 para mostrar um algoritmo polinomial para determiná-los quando o grafo pertence à algumas classe de grafos com poucos P_4 's.

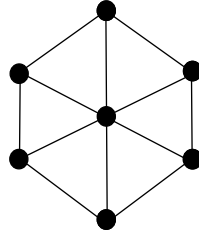
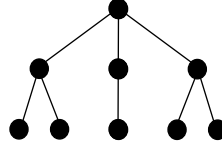
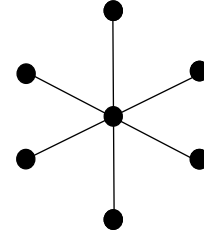
4.1 Introdução

Seja G um grafo finito. Dois jogadores, \mathcal{C} e \mathcal{R} , se revezam em uma partida na qual o conjunto de peças controladas pelo jogador \mathcal{C} (policiais) se movem entre vértices adjacentes de G e tentam capturar a peça controlada pelo jogador \mathcal{R} (ladrão). O jogo transcorre com informação perfeita, ou seja, cada jogador sabe a posição das peças de seu adversário nos vértices do grafo.

Na rodada 0, o jogador \mathcal{C} escolhe em quais vértices deseja posicionar seus policiais. Em seguida, o jogador \mathcal{R} , conhecendo as posições iniciais dos policiais, posiciona o ladrão. A partir de então, os jogadores se alternam, começando por \mathcal{C} , movimentando suas peças entre vértices adjacentes. Não é obrigatório que o jogador mova todas as suas peças em uma única rodada, apenas um subconjunto delas, podendo até mesmo “passar a rodada” sem mover qualquer uma de suas peças. Além disso, vários policiais podem ocupar o mesmo vértice simultaneamente. O jogo termina, com vitória dos policiais, se após um número finito de rodadas um policial ocupa o mesmo vértice do ladrão. Caso contrário, o ladrão vence.

Uma estratégia para o jogador \mathcal{C} em um grafo G é uma função que, dada a posição dos policiais e do ladrão após o movimento do ladrão, retorna para onde cada um dos policiais deve se mover em seguida. Analogamente, uma estratégia para o jogador \mathcal{R} é também uma função que, dada a posição dos policiais e do ladrão após a movimentação dos policiais, retorna para onde o ladrão deve se mover. Uma estratégia de um dos jogadores é dita vencedora se ao segui-la este sempre vence independentemente de como jogar seu adversário. Observe que, se um dos jogadores possuir uma estratégia vencedora em um grafo G , então seu adversário não pode possuir também uma estratégia vencedora para G , uma vez que se ambos utilizassem estas estratégias apenas um deles poderia vencer.

O jogo de polícia e ladrão descrito acima foi introduzido por Quilliot em sua tese de doutorado (QUILLIOT, 1978) e posteriormente, de maneira independente, por Nowakowski e Winkler (NOWAKOWSKI and WINKLER, 1983). Ambos os trabalhos consideraram que o jogador perseguidor controlava apenas uma “peça” (policial) no grafo visando a capturar a “peça” (ladrão) do seu oponente. Nesse caso, se \mathcal{C} possui uma estratégia vencedora dizemos que o grafo é *cop-win*.

Figura 16 – Exemplos de grafos *cop-win*Figura 17 – Grafo
com vértice universalFigura 18 –
ÁrvoreFigura 19 – Grafo
Estrela

Fonte: Elaborado pelo autor.

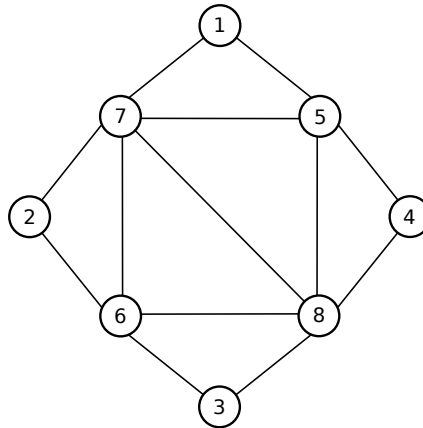
O trabalho de Quilliot, Nowakowski e Winkler se concentrou em caracterizar os grafos *cop-win*. Podemos observar facilmente que um grafo com um vértice universal é *cop-win*, bastando que \mathcal{C} escolha como posição inicial do seu policial o próprio vértice universal que, independentemente da posição inicial escolhida para o ladrão, \mathcal{C} vencerá em seu próximo movimento. Árvores também são grafos *cop-win*, neste caso a estratégia do jogador \mathcal{C} é movimentar seu policial em direção ao vértice ocupado pelo ladrão no único caminho que há entre eles.

Embora, árvores (Figuras 18 e 19) e grafos com vértices universais (Figura 17) sejam ambos *cop-win*, essas duas condições não caracterizam todos os grafos *cop-win*. Por exemplo o grafo da Figura 20 é *cop-win*, embora não seja uma árvore e não possua vértice universal. Para vencer, basta que o jogador \mathcal{C} posicione inicialmente seu único policial no vértice 8. Desta maneira, \mathcal{R} só poderia posicionar seu ladrão nos vértices 1 ou 2, caso contrário o ladrão seria posicionado em um vértice vizinho do ocupado pelo policial e seria capturado na próxima rodada. Independentemente se o ladrão fosse posicionado em 1 ou em 2, o policial deveria apenas se mover para 7 no seu próximo turno. Como $N[1] \subseteq N[7]$ e $N[2] \subseteq N[7]$ o ladrão seria inevitavelmente capturado no próximo movimento do policial.

Um grafo $G = (V, E)$ é dito *dismantable* se existe uma ordenação de V , $\{v_1, \dots, v_n\}$, tal que para todo $1 \leq i < n$, existe $j > i$ onde $N[v_i] \subseteq N[v_j]$ no grafo $G[\{v_i, v_{i+1}, \dots, v_n\}]$. Neste caso dizemos que v_j cobre v_i em $G[\{v_i, v_{i+1}, \dots, v_n\}]$. Nowakowski e Winkler provaram que um grafo é *cop-win* se, e somente se, ele é *dismantable* (NOWAKOWSKI and WINKLER, 1983).

Clarke e Nowakowski descreveram uma estratégia vencedora para um policial em um grafo *dismantable* utilizando a ordenação dos seus vértices (CLARKE and NOWAKOWSKI, 2001). Nessa estratégia no i -ésimo turno o policial joga como se estivesse no grafo $G[v_{n-i}, \dots, v_n]$ sempre perseguindo a “sombra do ladrão” neste grafo. A sombra do ladrão no grafo $G[\{v_{n-i}, \dots, v_n\}]$ é o vértice v_{j_k} tal que existe uma sequência $v_{j_1}, v_{j_2}, \dots, v_{j_k}$ na qual para todo $1 \leq l < k$ temos que $v_{j_{l+1}}$ cobre v_{j_l} , v_{j_1} é a atual posição do ladrão em G e v_{j_k} é o único vértice de $\{v_{n-i}, \dots, v_n\}$ na sequência. Tal estratégia garante a captura do ladrão em n rodadas.

Figura 20 – Exemplo de grafo *dismantable* com a ordenação dos vértices



Fonte: Elaborado pelo autor.

Aigner e Fromme deram início ao estudo da versão do jogo de polícia e ladrão em grafos onde o jogador \mathcal{C} pode controlar múltiplos policiais (AIGNER and FROMME, 1984). Neste caso, uma possível solução para garantir a vitória dos policiais seria para cada vértice do grafo (ou pelo menos para cada vértice de um conjunto dominante) utilizar um policial. No entanto, como já vimos nos exemplos acima, esse número pode ser muito maior que o necessário. Portanto, o objetivo do problema é, dado um grafo G , encontrar o menor número k tal que \mathcal{C} possui uma estratégia vencedora utilizando k policiais em G .

Dado um grafo G dizemos que G é k -cop-win se existe uma estratégia com k policiais para sempre capturar um ladrão em G . O *cop-number* de um grafo G , denotado por $c(G)$, é o menor inteiro k tal que G é k -cop-win.

Hahn e MacGillivray mostraram que se o número k de policiais é fixo, então podemos determinar se um grafo G é k -cop-win em tempo polinomial (HAHN and MACGILLIVRAY, 2006). Para tanto Hahn e MacGillivray mostraram como construir um digrafo D_G no qual cada vértice simbolizava uma configuração do jogo (posição dos policiais e do ladrão) em G e cada arco representava um movimento possível entre duas configurações. Tal digrafo possui $2|V(G)|^{k+1}$ vértices, considerando que para cada configuração há um vértice para o caso do próximo movimento ser dos policiais e outro caso seja a vez do ladrão se movimentar. A partir daí Hahn e MacGillivray mostraram que um simples algoritmo de busca aplicado a D_G é suficiente para determinar se k policiais serão bem sucedidos em capturar um fugitivo em G . No entanto, Kratochvíl et al. mostraram que, dado um inteiro k e um grafo G , determinar se $c(G) \leq k$ é um problema NP-difícil (FOMIN et al., 2010) utilizando uma redução para o problema do conjunto dominante. Não é conhecido se o problema pertence, ou não, à NP. Recentemente, Kinnersley mostrou que o problema é EXPTIME-completo (KINNERSLEY, 2015).

Além de sua importância teórica, devido a sua dificuldade, o jogo de polícia e ladrão possui diversas aplicações práticas como, por exemplo, simular um vírus em uma

rede sendo procurado por programas antivírus executados nos nós da rede, ou mesmo a busca por um fugitivo em um sistema de cavernas.

Um dos tópicos mais estudados sobre o *cop-number* é limitar seu valor por uma função do número de vértices do grafo. Seja n um inteiro positivo, $c(n)$ é o valor máximo que o *cop-number* pode assumir ao considerarmos todos os grafos com n vértices. Isto é, $c(n) = \max\{c(G) : |V(G)| = n\}$. Em 1985, H. Meyniel propôs a seguinte conjectura em uma comunicação privada com P. Frankl.

Conjectura 4.1 $c(n) = O(\sqrt{n})$.

A conjectura de Meyniel permanece como um dos maiores desafios no estudo de *cop-number*. O melhor limite conhecido foi obtido por Lu e Peng e estabelece que, $c(n) = O\left(\frac{n}{2^{1-o(n)}\sqrt{\log_2 n}}\right)$ (LU and PENG, 2012). Note que embora este seja o melhor limite conhecido para o *cop-number*, o resultado de Lu e Peng ainda está muito distante de $O(\sqrt{n})$.

Pralat mostrou uma família infinita de grafos, onde cada grafo G com n vértices tem *cop-number* $c(G) \geq \sqrt{\frac{n}{8}}$ e $c(G) \geq \sqrt{\frac{n}{8}} - n^{0.2625}$ para n suficientemente grande (PRALAT, 2010). O resultado de Pralat nos mostra que $c(n) = \Omega(\sqrt{n})$ nesta família de grafos. Desta maneira se a Conjectura de Meyniel for provada, este será o melhor limite possível para o *cop-number* em relação ao número de vértices do grafo.

Embora a Conjectura de Meyniel continue sem solução, podemos abordá-la para classes de grafos restritas, visando a encontrar um limite para o *cop-number* para tais classes. Aigner e Fromme mostraram que todo grafo planar possui *cop-number* no máximo 3 (AIGNER and FROMME, 1984). Andreae generalizou o resultado de Aigner e Fromme para grafos que não possuem K_5 como menor ou que não possuem $K_{3,3}$ como menor (ANDREAE, 1984) (os grafos planares são exatamente aqueles que não possuem nem K_5 nem $K_{3,3}$ como menor). Além disso, Andreae ainda provou que para qualquer grafo H , o *cop-number* da classe de grafos que não possuem H como menor é limitado pelo número de arestas de H (ANDREAE, 1986). Em 2012, Lu e Peng mostraram que se G é um grafo com diâmetro 2, ou um grafo bipartido com diâmetro no máximo 3, então $c(G) \leq 2\sqrt{n} - 1$ (LU and PENG, 2012). Neste capítulo iremos mostrar outras classes de grafos para as quais a Conjectura de Meyniel também é válida.

Um outro parâmetro relacionado ao jogo de polícia e ladrão em grafos é o chamado *capture time*, introduzido recentemente por A. Bonato et al. (BONATO *et al.*, 2009). A duração de um jogo é o número de rodadas (movimentação dos policiais seguida pelo movimento do ladrão), sem contar a rodada 0, realizadas até a captura do ladrão. O *k-capture time* de um grafo G , denotado por $\text{capt}_k(G)$, é a duração mínima de jogo com k policiais dado que o ladrão tenta impedir a captura pelo máximo tempo possível. Se $k = c(G)$ dizemos apenas *capture time* de G e denotamos por $\text{capt}(G)$.

O algoritmo de Hahn e MacGillivray para determinar se um número fixo de k policiais pode capturar um fugitivo em um dado grafo G também determina o $\text{capt}_k(G)$

em tempo polinomial (HAHN and MACGILLIVRAY, 2006). No entanto, Bonato et al. mostraram que o problema se torna NP-difícil para um número arbitrário de policiais (BONATO *et al.*, 2009).

Além de sua importância teórica, por pertencer à classe dos problemas NP-difíceis, determinar o *k-capture time* também tem importância prática. Considere, por exemplo, a aplicação do jogo de polícia e ladrão na qual um vírus (ladrão) se movimenta em uma rede de computadores. Embora possamos pensar em otimizar o número de programas antivírus (policiais) sendo executados nos nós da rede simultaneamente, também podemos imaginar que quanto mais tempo o vírus permanecer na rede sem ser exterminado, mais informações dessa rede serão comprometidas. Desta maneira pode ser interessante utilizar uma quantidade maior do que a necessária de policiais para capturar o vírus mais rapidamente. Podemos ver o *capture time*, como a versão temporal do *cop-number*.

Observe que se $k = |V(G)|$, então $\text{capt}_k(G) = 0$, uma vez que o jogador \mathcal{C} pode posicionar um policial em cada vértice de G e desta maneira impedir que \mathcal{R} posicione seu ladrão sem ser capturado. Consideraremos este um caso degenerado. Neste texto quando nos referirmos a $\text{capt}_k(G)$ teremos que $1 \leq k \leq |V(G)| - 1$.

Por ser relativamente recente, pouco se sabe sobre valores exatos ou limites para o *k-capture time*. Bonato et al. (BONATO *et al.*, 2009) mostraram, utilizando indução em $|V(G)|$, que se G é *cop-win* e $|V(G)| \geq 5$, então $\text{capt}(G) \leq |V(G)| - 3$. O limite em grafos *cop-win* foi melhorado para $\text{capt}(G) \leq |V(G)| - 4$ se $|V(G)| \geq 7$ (GAVENCIÁK, 2010), utilizando também indução em $|V(G)|$. Ademais, para todo $n \geq 7$, existe um grafo *cop-win* com n vértices e *capture time* $n - 4$ (GAVENCIÁK, 2010). Portanto este limite é ótimo. Mehrabian mostrou que, se G é o produto cartesiano de duas árvores, então $\text{capt}(G) = \lfloor \text{diam}(G)/2 \rfloor$, onde $\text{diam}(G)$ é o diâmetro do grafo G (MEHRABIAN, 2011). Recentemente, A. Bonato mostrou que o *capture time* do hipercubo de ordem n é $\Theta(n \log n)$ (BONATO *et al.*, 2013).

Neste capítulo mostramos algoritmos polinomiais para calcular o *cop-number* e o *k-capture time*, para $1 \leq k \leq |V(G)| - 1$, de grafos P_4 -tidy e $(q, q - 4)$ -grafos. Observe que calcular o *k-capture time* para todos estes valores de k é um problema NP-completo para grafos em geral, uma vez que k não é um valor fixo. Além disso mostramos que a conjectura de Meyniel é válida para grafos P_4 -tidy conexos e $(q, q - 4)$ -grafos conexos com pelo menos q vértices.

4.2 Parâmetros de captura em grafos com poucos P_4 's

Nesta seção iremos utilizar a decomposição dos grafos P_4 -tidy, como descrita no Teorema 2.3, para obter um algoritmo linear para calcular $c(G)$ e $\text{capt}_k(G)$ de qualquer grafo P_4 -tidy G . Para tanto iremos utilizar uma abordagem recursiva em que o número

cop-number e o *k-capture time* são ambos calculados para cada nó interno (que representa um subgrafo de G) da árvore de decomposição, baseados nos resultados obtidos para os seus dois filhos. As folhas da decomposição serão tratadas no teorema final desta seção. Repetimos este procedimento até calcular $c(G)$ e $\text{capt}_k(G)$.

O lema abaixo, apesar de simples, irá nos auxiliar a demonstrar outros resultados neste capítulo.

Lema 4.1 *Seja G um grafo não vazio com n vértices, então $\text{capt}_k(G) = 1$ se, e somente se, $\gamma(G) \leq k < n$.*

Prova: Suponha que o jogador \mathcal{C} possua $k \geq \gamma(G)$ policiais a sua disposição. Assim na rodada 0 \mathcal{C} posiciona cada policial em um vértice distinto de um conjunto dominante de G , os demais $k - \gamma(G)$ policiais podem ser posicionados em vértices quaisquer de G . Assim cada vértice será ocupado por um policial ou estará adjacente a um vértice ocupado por um policial. Logo, independente da posição inicial do ladrão, este será capturado na próxima rodada. Ademais, como G não é vazio e $k < n$, então há pelo menos um vértice em G que não é ocupado inicialmente por nenhum policial e portanto há uma posição “segura” para que o ladrão evite ser capturado ainda na rodada zero.

Agora suponha que $k < \gamma(G)$. Desta maneira, independentemente da forma como os policiais forem posicionados inicialmente pelo jogador \mathcal{C} , eles não podem ocupar um conjunto dominante de G . Ou seja, sempre irá existir um vértice de G no qual o ladrão pode se posicionar inicialmente de modo a não estar vizinho a nenhum vértice ocupado por um policial. Assim, qualquer policial levará pelo menos dois turnos para capturar o ladrão se ele permanecer parado e, portanto, $\text{capt}_k(G) > 1$. \square

Agora iremos iniciar o estudo dos nós internos da decomposição dos grafos P_4 -tidy calculando $c(G)$ e $\text{capt}_k(G)$ quando G é a união de dois grafos deste tipo.

Lema 4.2 *Seja $G = G_1 \cup G_2$, então:*

$$(a) \ c(G) = c(G_1) + c(G_2);$$

$$(b) \ \text{capt}(G) = \max\{\text{capt}(G_1), \text{capt}(G_2)\};$$

$$(c) \ \text{capt}_k(G) = \min_{\{(k_1, k_2) | k_1 + k_2 = k\}} \{\max\{\text{capt}_{k_1}(G_1), \text{capt}_{k_2}(G_2)\}\}.$$

Prova: (a) Certamente $c(G) \geq c(G_1) + c(G_2)$, uma vez que se tivermos menos que $c(G_1) + c(G_2)$ policiais, então o jogador \mathcal{C} deverá posicionar, na rodada 0, menos que $c(G_1)$ policiais em G_1 ou menos que $c(G_2)$ policiais em G_2 . Como não é possível para \mathcal{C} remanejar policiais de G_1 para G_2 ou vice versa basta que \mathcal{R} posicione o ladrão na componente com policiais insuficientes para evitar a captura indefinidamente.

Para mostrar que $c(G) \leq c(G_1) + c(G_2)$ basta observar que, se na rodada 0 o jogador \mathcal{C} distribuir $c(G_1)$ policiais em G_1 e $c(G_2)$ policiais em G_2 , \mathcal{C} pode usar uma estratégia vencedora para capturá o ladrão em G_1 ou em G_2 , independente de qual grafo será escolhido pelo ladrão para se posicionar inicialmente. Assim, garantimos a captura em um número finito de turnos.

(b) Dado que $c(G) = c(G_1) + c(G_2)$, como provado no item (a), temos que

a única maneira de \mathcal{C} garantir a captura do ladrão com este número de policiais é, na rodada 0, posicionar $c(G_1)$ policiais em G_1 e $c(G_2)$ policiais em G_2 . Assim se \mathcal{R} escolher posicionar o ladrão em algum vértice de G_1 este será capturado em $\text{capt}(G_1)$, caso \mathcal{R} escolha posicionar o ladrão em G_2 ele será capturado em $\text{capt}(G_2)$. Como \mathcal{R} faz as escolhas que resultam no pior caso para o jogador \mathcal{C} temos que \mathcal{R} posicionará o ladrão de maneira a fazer o jogo demorar o maior tempo possível, assim sendo, $\text{capt}(G) = \max\{\text{capt}(G_1), \text{capt}(G_2)\}$.

(c) Dado que o jogador \mathcal{C} posiciona, inicialmente, k_1 policiais em G_1 e k_2 policiais em G_2 , com um raciocínio análogo ao do item anterior podemos concluir que \mathcal{R} irá posicionar seu ladrão, na rodada 0, no grafo que maximiza $\{\text{capt}_{k_1}(G_1), \text{capt}_{k_2}(G_2)\}$. Desta maneira, para atingir o menor tempo de captura possível com k policiais, o jogador \mathcal{C} deve escolher os pares k_1 e k_2 tal que $k = k_1 + k_2$ de maneira a minimizar $\max\{\text{capt}_{k_1}(G_1), \text{capt}_{k_2}(G_2)\}$. \square

Lema 4.3 *Seja $G = G_1 \vee G_2$, então:*

$$(a) \ c(G) = \min\{2, c(G_1), c(G_2)\};$$

$$(b) \ \text{capt}(G) = \begin{cases} 1, & \text{se } c(G) = 2 \\ \min\{\text{capt}_1(G_1), \text{capt}_1(G_2)\}, & \text{se } c(G) = 1 \end{cases}$$

$$(c) \ \text{capt}_k(G) = \begin{cases} 1, & \text{se } k \geq 2 \\ \min\{\text{capt}_1(G_1), \text{capt}_1(G_2)\}, & \text{se } k = 1 \end{cases}$$

Prova: (a) Claramente $c(G) \leq 2$ uma vez que o jogador \mathcal{C} pode posicionar inicialmente um policial em G_1 e outro policial em G_2 e, desta maneira, tornar todo vértice de G adjacente a pelo menos um vértice ocupado por um policial. Assim, independente da posição inicial do ladrão ele será capturado na primeira rodada.

Suponha que $c(G_1) > 1$ e $c(G_2) > 1$, então podemos mostrar uma estratégia para o ladrão fugir indefinidamente de um policial em G , ou seja $c(G) > 1$. Suponha, sem perda de generalidade, que \mathcal{C} escolha posicionar seu único policial em um vértice de G_1 . Como $c(G_1) > 1$ o ladrão pode usar uma estratégia vencedora para se posicionar inicialmente em um vértice de G_1 e nunca ser capturado enquanto o policial permanecesse se movimentando apenas em G_1 . Caso o policial se movimente em algum momento para um vértice de G_2 , como todo vértice de G_1 (inclusive o atualmente ocupado pelo ladrão) é vizinho de todo vértice de G_2 e $c(G_2) > 1$, o ladrão pode adaptar sua estratégia vencedora em G_2 para se posicionar de forma a também evitar ser capturado enquanto o policial permanecesse se movimentando apenas em G_2 . Se o policial se movimentar novamente para um vértice de G_1 o processo se reinicia. Portanto, para que $c(G) = 1$ é necessário que $c(G_1) = 1$ ou $c(G_2) = 1$.

Se $c(G_1) = 1$, então \mathcal{C} pode posicionar inicialmente um policial em G_1 impedindo o jogador \mathcal{R} de escolher qualquer vértice de G_2 , uma vez que todos são adjacentes ao vértice com o policial. Desta maneira o jogo fica restrito ao grafo G_1 . Como $c(G_1) = 1$

o jogador \mathcal{C} captura o ladrão em um número finito de rodadas. O raciocínio é análogo no caso de $c(G_2) = 1$. Assim temos que $c(G) = \min\{2, c(G_1), c(G_2)\}$.

(b) Se $c(G) = 2$ então o jogador \mathcal{C} pode posicionar inicialmente um policial em G_1 e outro policial em G_2 e, desta maneira, tornar todo vértice de G adjacente a pelo menos um vértice ocupado por um policial. Assim, independente da posição inicial do ladrão ele será capturado na primeira rodada.

Se $c(G) = 1$ então $c(G_1) = 1$ ou $c(G_2) = 1$. Assim, seguindo um raciocínio análogo ao apresentado no item (a) quando $c(G_1) = 1$, basta que o jogador \mathcal{C} posicione um policial no grafo que minimiza $\{capt_1(G_1), capt_1(G_2)\}$ para obter o menor tempo de captura possível usando um policial.

(c) Se $k \geq 2$ então temos uma situação análoga à do item (b) quando $c(G) = 2$ e o ladrão é capturado na primeira rodada.

Se $k = 1$ então temos uma situação análoga à do item (b) quando $c(G) = 1$ e o ladrão é capturado em $\min\{capt_1(G_1), capt_1(G_2)\}$ rodadas. Observe que este valor pode ser infinito se os grafos G_1 e G_2 possuírem *cop-number* superior a 1. \square

Lema 4.4 *Seja G uma aranha com partição (R, C, S) , então:*

(a) $c(G) = 1$;

(b) $capt(G) = 2$;

(c) $capt_k(G) = \begin{cases} 2, & \text{se } G \text{ é uma aranha magra e } k < |C| \text{ ou } G \text{ é uma aranha gorda e } k = 1 \\ 1, & \text{se } G \text{ é uma aranha magra e } k \geq |C| \text{ ou } G \text{ é uma aranha gorda e } k \geq 2 \end{cases}$

Prova: (a) Se G é uma aranha magra então \mathcal{C} escolhe posicionar seu policial inicialmente no vértice $c_1 \in C$. Os únicos vértices de G que não são adjacentes a c_1 são os vértices s_i com $2 \leq i \leq |C|$, sem perda de generalidade podemos supor que \mathcal{R} escolhe o vértice s_2 como posição inicial do ladrão. Assim na próxima rodada \mathcal{C} moveria seu policial para c_2 e \mathcal{R} manteria o ladrão em s_2 pois seu único vizinho está ocupado por um policial. Observe também que neste caso o ladrão seria capturado na segunda rodada.

Se G é uma aranha gorda então \mathcal{C} escolhe posicionar seu policial inicialmente no vértice $c_1 \in C$. O único vértice de G que não é adjacente a c_1 é o vértice s_1 e, sendo assim, será escolhido por \mathcal{R} como posição inicial do ladrão. Na primeira rodada o jogador \mathcal{C} escolhe o vértice c_2 que é adjacente a s_1 . Logo o jogador \mathcal{R} deve mover o ladrão se desejar evitar a captura, no entanto todos os vizinhos de s_1 pertencem a C e, portanto, são também adjacentes a c_2 . Assim o ladrão será inevitavelmente capturado na segunda rodada.

(b) Diretamente do item anterior temos que $capt(G) = 2$.

(c) Se G é uma aranha magra já mostramos que para qualquer $k \geq 1$ temos que $capt_k(G) \leq 2$. Ademais pelo Lema 4.1 $capt_k(G) = 1$ apenas quando $k \geq \gamma(G)$. Como as patas de uma aranha magra são vértices com apenas um vizinho, todo conjunto dominante de G deve conter pelo menos cada pata ou seu vizinho na clique e, portanto, $\gamma(G) \geq |S| = |C|$. Além disso, como todo vértice de R é adjacente a todo vértice de C

em G , temos que C é um conjunto dominante mínimo de G .

Se G é uma aranha gorda já mostramos que $\text{capt}_1(G) = 2$. Para completarmos a prova basta mostrarmos que $\text{capt}_2(G) = 1$. Para tanto basta observar que dois vértices distintos de C formam um conjunto dominante em G . \square

No Lema abaixo iremos assumir que G não é uma aranha, uma vez que este caso já foi tratado no Lema anterior.

Lema 4.5 *Seja G uma quase-aranha com partição (R, C, S) , se G é uma quase-aranha magra, então:*

$$(a) \ c(G) = \begin{cases} 2, & \text{se o vértice duplicado pertence a } C \text{ e forma um } \overline{K_2} \text{ com sua cópia.} \\ 1, & \text{caso contrário.} \end{cases}$$

$$(b) \ \text{capt}_k(G) = \begin{cases} 1, & \text{se } k \geq |C|. \\ 1, & \text{se } k \geq |C| - 1 \text{ e o vértice duplicado pertence a } C \text{ e forma um} \\ & K_2 \text{ com sua cópia.} \\ 2, & \text{caso contrário.} \end{cases}$$

Se G é uma quase-aranha gorda, então $c(G) = 1$ e:

$$(c) \ \text{capt}_k(G) = \begin{cases} 2, & \text{se } k = 1. \\ 1, & \text{se } k > 1. \end{cases}$$

Prova: (a) Suponha que G é uma quase-aranha magra cujo vértice duplicado c_i pertence a C e forma um $\overline{K_2}$ com sua cópia c'_i . Iremos mostrar uma estratégia vencedora para o ladrão jogando contra um policial em G . A estratégia do ladrão depende apenas da posição do policial. Se o policial estiver em c_i o ladrão irá se posicionar em c'_i e vice versa. Se o policial estiver em s_i deve se posicionar em um vértice de C diferente de c_i e c'_i . Se o policial estiver em qualquer outro vértice do grafo o ladrão deve se posicionar em s_i . Como o ladrão pode inicialmente escolher sua posição inicial de maneira a se adequar a esta estratégia, e mantém o policial sempre a uma distância pelo menos 2 de si ao executá-la. Temos então que $c(G) > 1$.

No entanto se o ladrão estiver enfrentando dois policiais estes conseguem capturá-lo em no máximo dois turnos. Basta que os policiais se posicionem inicialmente em c_i e c'_i , assim se o ladrão se posicionar em s_i ou em qualquer vértice de $C \cup R \setminus \{c_i, c'_i\}$ ele será capturado no próximo turno. Desta maneira o ladrão deve se posicionar inicialmente em algum vértice $s_j \in S \setminus \{s_i\}$. Se o fizer, para que os policiais o capturem, basta que um deles se movimente para c_j , como $N[s_j] \subseteq N[c_j]$, o ladrão não tem como fugir e será capturado no turno seguinte.

Suponha que G é uma quase-aranha magra cujo vértice duplicado pertence a S ou forma um K_2 com sua cópia. Podemos mostrar uma estratégia vencedora para um policial capturar o ladrão em dois turnos. O policial deve escolher como posição inicial um vértice c_i de C , assim se o ladrão se posicionar em s_i (e possivelmente s'_i) ou em qualquer vértice de $C \cup R$ ele será capturado no próximo turno. Assim o ladrão deve escolher um

vértice s_j (possivelmente s'_j) de S diferente de s_i (e possivelmente s'_i). Basta agora que o policial se movimente para c_j para então capturar o ladrão no turno seguinte.

(b) Pelo Lema 4.1 temos que $\text{capt}_k(G) = 1$ apenas quando $k \geq \gamma(G)$. De maneira semelhante ao que fizemos no Lema 4.4 podemos observar que quando o vértice duplicado não pertence a C ou forma um $\overline{K_2}$ com sua cópia, C é um conjunto dominante mínimo de G , ou seja, $\gamma(G) = |C|$. Caso contrário, se o vértice duplicado pertencer a C e formar um K_2 com sua cópia, temos um conjunto dominante mínimo com $|C| - 1$ vértices (todos os vértices de C exceto a cópia). Para quaisquer valores de k inferiores a $|C|$ (ou $|C| - 1$, dependendo do vértice duplicado), pelas estratégias mostradas no item (a) temos que $\text{capt}_k(G) = 2$.

(c) Como uma aranha gorda com $|C| = 1$ é desconexa e com $|C| = 2$ é também uma aranha magra, podemos assumir que uma quase-aranha gorda possui pelo menos 3 vértices originais em C (sem contar um possível vértice duplicado). Assim inicialmente o policial pode se posicionar em um vértice não duplicado c_j de C . Observe que se posicionando desta maneira o policial fica adjacente a todos os vértices de G exceto s_j (e sua possível cópia). Desta maneira o ladrão deve escolher s_j (ou sua cópia) como sua posição inicial. Basta agora que o policial se mova para outro vértice não duplicado de C (isto é possível uma vez que C possui pelo menos 3 vértices “originais”), encurralando o ladrão que, não tendo para onde fugir, será capturado no turno seguinte.

Para mostrar que $\text{capt}_k(G) = 1$ quando $k \geq 2$, basta utilizar o Lema 4.1 e observar que quaisquer dois vértices “originais” de C formam um conjunto dominante em G . \square

Com os resultados obtidos acima podemos obter um algoritmo linear para calcular $c(G)$ e $\text{capt}_k(G)$ para os grafos P_4 -tidy.

Teorema 4.1 *Dado um grafo P_4 -tidy G podemos determinar $c(G)$ e $\text{capt}_k(G)$ em tempo $O(n + m)$. Ademais para todo grafo P_4 -tidy conexo G , $c(G) \leq 2$.*

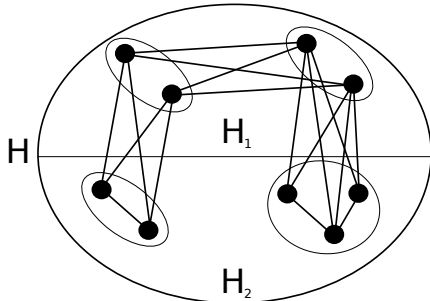
Prova: Primeiro, utilizando o Teorema 2.3, calculamos a árvore de decomposição de G , T_G , em tempo $O(m + n)$. Cada folha G' de T_G é um K_1 , C_5 , P_5 , $\overline{P_5}$ ou uma quase-aranha sem cabeça. No caso de G' ser um grafo quase-aranha sem cabeça, podemos calcular em tempo constante utilizando os Lemas 4.4 e 4.5. Os casos em que G' é um K_1 , C_5 , P_5 , $\overline{P_5}$ são trivialmente solucionados.

Cada vértice interno G' de T_G é um subgrafo grafo de G resultante de operações (\vee , \cup ou \uplus) sobre os seus filhos. Assim, se G' é o resultado de uma operação \cup podemos calcular $c(G)$ e $\text{capt}_k(G)$ em tempo constante (assumindo que os filhos desde já foram resolvidos) utilizando o Lema 4.2. Se G' é o resultado de uma operação \vee podemos calcular $c(G)$ e $\text{capt}_k(G)$ em tempo constante utilizando o Lema 4.3. Se G' é o resultado de uma operação \uplus (formando assim uma quase-aranha) podemos calcular $c(G)$ e $\text{capt}_k(G)$ em tempo constante utilizando os Lemas 4.4 e 4.5.

Como T_G possui $O(n)$ vértices, todo o processo, até chegar na raiz G de T_G ,

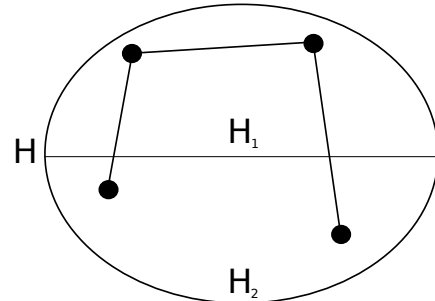
Figura 21 – Exemplo de p-componente separável (módulos em destaque) e sua respectiva p-componente característica

Figura 22 – p-componente separável



Fonte: Elaborado pelo autor.

Figura 23 – p-componente característica



Fonte: Elaborado pelo autor.

leva tempo $O(n + m)$. Ademais para todos os casos acima temos que $c(G) \leq 2$. \square

Observe que o Teorema 4.1 mostra que a conjectura de Meyniel é válida para grafos P_4 -tidy conexos com pelo menos 4 vértices. Ademais todos os grafos conexos com no máximo 3 vértices são *cop-win*. Portanto a Conjectura de Meyniel é válida para todos os grafos P_4 -tidy conexos.

Os Lemas anteriores também podem ser utilizados para obter um algoritmo polinomial para calcular o *cop-number* e o *k-capture time* de qualquer $(q, q - 4)$ -grafo com q fixo. Para tanto iremos utilizar a *decomposição homogênea* de grafos introduzida por Jamison e Olariu (JAMISON and OLARIU, 1995).

Um *módulo* em um grafo G é um conjunto de vértices indistinguíveis para os demais vértices de G . Isto é, um conjunto de vértices M é um módulo de um grafo G se para cada vértice $v \in V(G) \setminus M$, tem-se que ou v é adjacente a todos os vértices de M ou v é adjacente a nenhum vértice de M .

Jamison e Olariu examinaram o processo de compactar cada módulo de uma p-componente separável H com separação (H_1, H_2) até um único vértice mantendo as adjacências entre os vértices característicos dos módulos. Eles mostraram que o grafo obtido desta maneira, ao qual eles chamaram de *p-componente característica* de H , é um grafo split, onde os vértices da clique representam os módulos em H_1 e os vértices do conjunto independente representam os módulos em H_2 (JAMISON and OLARIU, 1995). Ademais, a p-componente característica pode ser obtida em tempo polinomial no tamanho de H . Abaixo temos um exemplo de p-componente separável e sua respectiva p-componente característica.

Teorema 4.2 *Dado um $(q, q - 4)$ -grafo podemos determinar $c(G)$ e $\text{capt}_k(G)$ em tempo polinomial, com q fixo. Ademais para todo $(q, q - 4)$ -grafo conexo G com pelo menos q vértices, $c(G) \leq 2$.*

Prova: Se G for desconexo basta calcularmos os valores de $c(G)$ e $\text{capt}_k(G)$ para cada uma das componentes de G . Assumindo que G é conexo, utilizando o Teorema 2.2,

calculamos a árvore de decomposição de G , T_G , em tempo $O(m+n)$. Os casos nos quais a raiz de T_G é a junção de dois grafos ou uma aranha são tratados pelos Lemas 4.3 e 4.4. Os quais garantem que $c(G) \leq 2$.

Assuma agora que a raiz da árvore T_G for gerada por uma operação $G' \uplus H$, onde $H = (H_1, H_2)$ é uma p -componente separável com menos que q vértices. Então podemos usar a p -componente característica de H para mostrar que $c(G) \leq 2$. Iniciamos o jogo colocando dois policiais c_1 e c_2 em vértices de módulos distintos de H_1 . Como os módulos de H_1 formam uma clique, então esses policiais estão adjacentes a todos os vértices de H_1 e todos os vértices de G' e alguns vértices de H_2 . Assim o ladrão não possui outra opção para sua posição inicial senão escolher algum vértice de H_2 não adjacente a ambos os policiais.

No próximo turno o policial c_1 se move para um vértice de H_1 adjacente ao vértice escolhido inicialmente pelo ladrão. Observe que, como os módulos de H_2 formam um conjunto independente na p -componente característica de H , então o ladrão só pode se mover para um vértice de H_1 ou para um vértice de H_2 pertencente ao mesmo módulo que se encontra agora. Caso escolha se manter no mesmo módulo o ladrão será capturado no próximo turno por c_1 . Caso vá para um vértice de H_1 , então c_1 ou c_2 irá capturá-lo no próximo turno.

Acima garantimos não apenas que $c(G) \leq 2$ como também que $\text{capt}_k(G) \leq 2$ para todo $k \geq 2$, uma vez que incluir mais policiais não pode adiar a captura. Para determinar se G é *cop-win* podemos utilizar o algoritmo polinomial de Hahn e MacGillivray (HAHN and MACGILLIVRAY, 2006) e, em caso afirmativo, ao mesmo tempo calcular o $\text{capt}_1(G)$. Ademais, $\text{capt}_k(G) = 1$ se e somente se $k \geq \gamma(G)$. Logo, para determinarmos o valor de $\text{capt}_k(G)$, para $2 \leq k \leq |V(G)| - 1$ basta verificar se $k \geq \gamma(G)$, o que também pode ser feito em tempo polinomial, uma vez que só os vértices de H precisam ser analisados e H possui menos que q vértices.

Resta-nos mostrar o caso em que G possui menos que q vértices. Neste caso, novamente, executamos o algoritmo de Hahn e MacGillivray (HAHN and MACGILLIVRAY, 2006) para cada quantidade de policiais variando entre $1, \dots, q$. Como q é um valor fixo, tal procedimento toma tempo polinomial. \square

O Teorema 4.2 implica que a Conjectura de Meyniel é válida para $(q, q-4)$ -grafos com quantidade grande o suficiente de vértices.

Corolário 4.1 *A conjectura de Meyniel é válida para todo $(q, q-4)$ -grafo conexo G com pelo menos q vértices.*

5 JOGO DO ESPIÃO

No capítulo anterior estudamos um dos mais famosos jogos de perseguição em grafos. Há diversas adaptações deste tipo de jogo para as mais diversas situações. Neste capítulo estudaremos um outro jogo de perseguição que introduzimos recentemente, o *Jogo do Espião* (COHEN *et al.*, 2016).

5.1 Introdução

Neste jogo, dado um grafo G e inteiros $k > 0$, $s > 0$ e $d \geq 0$, o jogador \mathcal{G} controla um conjunto de k guardas enquanto o jogador \mathcal{S} controla um espião. Inicialmente o jogador \mathcal{S} posiciona seu espião em um vértice de G e depois o jogador \mathcal{G} posiciona seus k guardas. A partir de então os jogadores se revezam, começando por \mathcal{S} , a mover suas respectivas peças entre os vértices de G . O espião, em uma jogada, pode se mover para qualquer vértice a uma distância no máximo s (sua velocidade) de sua posição atual, enquanto cada guarda só pode mover-se para vértices à distância no máximo 1, isto é cada policial tem velocidade 1. O espião pode utilizar um caminho P (com tamanho no máximo s) para se movimentar mesmo que vértices de P estejam ocupados por guardas. De fato, diferente do que ocorre no jogo de polícia e ladrão, o espião e os guardas podem até mesmo ocupar o mesmo vértice sem que isso determine o fim da partida. Além disso, cada guarda e o espião podem optar por permanecer no mesmo vértice em que estão, isto é, podem escolher “pular” sua etapa de movimentação. Assim como ocorre no Jogo de Polícia e Ladrão, o Jogo do Espião transcorre com informação perfeita, isto é, ambos os jogadores conhecem as posições e movimentos anteriores do seu adversário.

O espião deseja se comunicar com seu país de origem e repassar os segredos que descobriu. No entanto, ele só pode fazê-lo se estiver a uma distância maior que d de qualquer vértice ocupado por um guarda ao final da etapa de movimentação dos guardas. O espião vence se após um número finito de turnos conseguir se comunicar com seu país. Os guardas vencem se jamais permitem que o espião se comunique. Observe que, diferentemente do que ocorria no jogo de polícia e ladrão, no jogo do espião o objetivo dos guardas não é capturar o espião, mas sim perseguí-lo indefinidamente.

Uma estratégia para o jogador \mathcal{G} em um grafo G é uma função que, dada a posição dos guardas e do espião após o movimento do espião, retorna para onde cada um dos guardas deve se mover em seguida. Analogamente, uma estratégia para o jogador \mathcal{S} é também uma função que, dada a posição dos guardas e do espião após a movimentação dos guardas, retorna para onde o espião deve se mover. Uma estratégia de um dos jogadores é dita vencedora se ao segui-la este sempre vence independentemente de como jogar seu adversário. Observe que, se um dos jogadores possuir uma estratégia vencedora em um grafo G , então seu adversário não pode possuir também uma estratégia vencedora para

G , uma vez que se ambos utilizassem estas estratégias apenas um deles poderia vencer.

Assim como no Jogo de Polícia e Ladrão desejamos saber o menor número possível de policiais para vencer o jogo, no Jogo do Espião desejamos determinar o menor número de guardas. Dado um grafo G e inteiros $s > 0$ e $d \geq 0$, o (s, d) -guard-number de G , $g_d^s(G)$, é o menor número de guardas necessários para vencer o Jogo do Espião em G onde s é a velocidade do espião e d é a distância máxima que um guarda pode vigiar.

Podemos definir também uma noção dual de $g_d^s(G)$. Dado um grafo G e inteiros $k > 0$, $s > 0$ então $d_k^s(G)$ é a menor distância d tal que k guardas conseguem impedir que um espião com velocidade s se comunique, mantendo-o sempre a uma distância inferior a d de pelo menos um guarda a cada turno.

Podemos definir ainda uma versão do jogo do espião na qual os guardas são posicionados primeiro. No entanto, o espião poderia ser posicionado à uma distância maior que d de todos os guardas já no primeiro turno. Neste caso, temos que mudar as condições de vitória dos jogadores para o seguinte: o jogador \mathcal{G} vence se, após um número finito de rodadas, ele consegue manter pelo menos um guarda a distância no máximo d do espião a cada turno, caso contrário \mathcal{S} vence. Isto é, o espião precisa se comunicar indefinidamente para vencer, enquanto os guardas precisam garantir que o espião se comunique no máximo um número finito de vezes.

Na maior parte deste capítulo estudaremos a versão do jogo em que o espião é posicionado antes dos guardas, no entanto mostramos abaixo que ambas as versões estão fortemente relacionadas. Nas seguintes proposições, considere um espião com velocidade s tentando se distanciar pelo menos $d + 1$ arestas em relação aos guardas em um grafo G .

Proposição 5.1 *Se o espião vence o jogo quando é posicionado primeiro, então ele vence o jogo quando é posicionado após os guardas.*

Prova: Se \mathcal{S} vence ao posicionar seu espião primeiro, então existe um vértice v_0 tal que a partir de v_0 o espião possui uma estratégia vencedora para alcançar um vértice a uma distância superior a d de qualquer guarda independente da posição inicial dos guardas e da estratégia adotada por \mathcal{G} . Assim na versão em que é posicionado após os guardas a estratégia do espião é, a partir de v_0 , utilizar a estratégia anterior para poder se comunicar uma vez com seu país e após fazê-lo retornar à v_0 para reiniciar o processo, repetindo-o indefinidamente. \square

Embora, como mostraremos mais adiante, o inverso não seja sempre verdade, podemos calcular um limite para o número de guardas necessários para defender o grafo G utilizando o *cop-number* de G , $c(G)$.

Proposição 5.2 *Se k guardas vencem o jogo quando o espião é posicionado primeiro, então $k + c(G) - 1$ guardas vencem o jogo quando o espião é posicionado após os guardas.*

Prova: Assuma que k guardas possuem uma estratégia vencedora \mathcal{E} quando o espião é posicionado primeiro. Tal estratégia é descrita da seguinte forma. Para todo vértice $v \in V(G)$ no qual o espião estiver posicionado, o guarda g_i , com $1 \leq i \leq k$, deve

ser posicionado no vértice $pos(i, v)$, tal que para todo vértice $w \in V(G)$ a distância no máximo s de v (todos os vértices que podem ser alcançados pelo ladrão no seu próximo movimento) temos que $pos(i, w) \in N[pos(i, v)]$. Ademais para todo $v \in V(G)$ existe $1 \leq j \leq k$ tal que $pos(j, v)$ está à distância no máximo d de v .

Assuma então que $k + c(G) - 1$ guardas são posicionados primeiro. Devemos mostrar que, após um número finito de turnos, se o espião estiver em um vértice $v \in V(G)$, então para todo $1 \leq i \leq k$ os vértices $pos(i, v)$ estarão ocupados por guardas que, a partir de então, podem usar a estratégia \mathcal{E} para impedir que o espião se comunique.

Em um dado momento do jogo onde o espião está posicionado no vértice v , seja $0 \leq j < k$ tal que para todo $0 \leq i \leq j$ os vértices $pos(i, v)$ estão todos ocupados por guardas (caso $j = 0$ nenhuma das k posições estão ocupadas). Para cada $0 \leq i \leq j$ um guarda posicionado no vértice $pos(i, v)$ seguirá a estratégia \mathcal{E} até o final do jogo, os demais $k + c(G) - 1 - j \geq c(G)$ guardas irão tentar “capturar” a posição $pos(j + 1, v)$ (que irá ser modificada a cada turno como um ladrão com velocidade 1 em G). Como pelo menos $c(G)$ guardas livres irão perseguir tal posição, então em um número finito de turnos um guarda irá ocupar a posição $pos(j + 1, w)$ (onde w é a posição do espião neste turno).

Repetindo o processo um número finito de vezes (no máximo k) temos que, em um dado momento, o espião estará em um vértice v e os vértices $pos(i, v)$ ($1 \leq i \leq k$) estarão todos ocupados por pelo menos um guarda, o que conclui a prova. \square

O limite obtido na Proposição 5.2 é apertado quando consideramos $s = 1$ e $d = 0$. De fato, para todo grafo G temos que $g_0^1(G) = 1$, pois basta colocar um guarda na mesma posição inicial do espião e segui-lo. No entanto, se os guardas são colocados primeiro, o jogo se torna equivalente ao jogo de polícia e ladrão, visto que os guardas devem primeiro alcançar o espião e depois segui-lo aonde quer que ele vá. Portanto, no segundo caso, são necessários pelo menos $c(G)$ guardas.

Em 2010 Bonato et al. definiram o jogo de polícia e ladrão com raio de captura d (BONATO, CHINIFOROOSHAN, and PRALAT, 2010), no qual o ladrão é capturado se em qualquer momento do jogo um policial é posicionado a uma distância pelo menos d do vértice ocupado pelo ladrão. Observe que, para qualquer distância d , o jogo de polícia e ladrão com raio de captura d é equivalente à versão do jogo do espião em que os guardas são posicionados primeiro com parâmetros $(s = 1, d)$. Isso ocorre pois no momento em que o espião fica a uma distância d pela primeira vez, como os guardas são tão rápidos quanto ele, ele jamais conseguirá se afastar novamente para se comunicar.

A equivalência dos dois jogos, no entanto, não se mantém se aumentarmos a velocidade dos fugitivos em ambos. Um policial é suficiente para capturar um ladrão em uma árvore, independente da velocidade do mesmo ou do raio de captura, no entanto, como mostraremos na Seção 5.4, se d é suficientemente pequeno em relação a n , então são necessários $O(n)$ guardas para impedir o espião de se comunicar em um caminho com n

vértices. Isso acontece pelo fato do espião poder cruzar ou até ocupar o mesmo vértice de um guarda e, como possui velocidade superior a de seus perseguidores, poder se distanciar rapidamente.

Um *conjunto d -dominante* em um grafo $G = (V, E)$ é um subconjunto de vértices D tal que todo $v \in V$ está a distância no máximo d de um vértice de D (HENNING, SWART, and SWART, 1991; SLATER, 1976). Seja $\gamma_d(G)$ a mínima cardinalidade de um conjunto d -dominante em G . Certamente $g_d^s(G) \leq \gamma_d(G)$ para todo $s > 0$, uma vez que bastaria manter um guarda em cada vértice do conjunto d -dominante para impedir o espião de se comunicar.

No entanto $\gamma_d(G)$ pode diferir tanto quanto se queira de $g_d^s(G)$ para todo $s > 0$, $d \geq 0$. Seja G o grafo composto por um ciclo C com n vértices e um vértice x tal que para todo $v \in C$ temos um caminho, disjunto dos demais exceto pelo vértice x , de tamanho $d + 1$ entre x e v . No grafo G descrito acima temos que $g_d^s(G) = 2$ (dois guardas se revezando entre x e um de seus vizinhos) enquanto $\gamma_d(G) = \Omega(n)$.

No *jogo de dominação eterna* (GODDARD, HEDETNIEMI, and HEDETNIEMI, 2005; GOLDWASSER and KLOSTERMEYER, 2008; KLOSTERMEYER and MACGILLIVRAY, 2009; KLOSTERMEYER and MYNHARDT, 2011) em um grafo G , dado uma distância fixa d , um conjunto de k defensores ocupam vértices do grafo. A cada turno o jogador atacante seleciona um vértice $v \in V(G)$ para ser atacado. Em seguida, cada defensor pode se mover para um vértice adjacente. Um defensor consegue proteger todos os vértices a distância no máximo d . O jogador atacante vence se em qualquer turno ele conseguir atacar um vértice que não pode ser protegido, caso contrário o jogador defensor vence. O jogo do espião, quando o espião tem velocidade ilimitada (pelo menos o diâmetro do grafo), é equivalente ao jogo de dominação eterna.

Os resultados que mostram a dificuldade computacional do problema descritos neste capítulo, bem como o fato do jogo do espião generalizar vários outros problemas de perseguição já estabelecidos na literatura (jogo de polícia e ladrão, Jogo de polícia e ladrão com raio de captura e o jogo de dominação eterna), comprovam a importância teórica do jogo do espião.

Na Seção 5.2 mostramos que decidir se $g_d^s(G) \leq k$ é NP-difícil e inaproximável em tempo polinomial por um determinado fator (a menos que $P = NP$). Na Seção 5.3 mostramos que o problema do jogo do espião em grafos direcionados é PSPACE-difícil na versão em que os guardas são posicionados antes do espião. Na Seção 5.4 mostramos limites para $d_k^s(G)$ quando G é um caminho ou um ciclo. Na seção 5.5 apresentamos um algoritmo linear para calcular $g_d^s(G)$ para quaisquer $s > 0$ e $d \geq 0$, quando o grafo G é um grafo P_4 -tidy.

Parte dos resultados deste capítulo foram publicadas em (COHEN *et al.*, 2018).

5.2 NP-Compleitude e Inaproximabilidade

Nesta seção, provamos que decidir se $g_d^s(G) \leq k$ é NP-difícil para todo $s, d \geq 0$. A prova será dividida em três casos: (i) $s \geq 2d+2$, (ii) $2d+2 > s \geq d+2$ e (iii) $s \leq d+1$.

As reduções são obtidas a partir do problema Cobertura de Conjuntos que é um dos famosos 21 problemas NP-completos de Karp (KARP, 1972). Uma instância do problema de Cobertura de Conjuntos é uma família $\mathcal{S} = \{S_1, \dots, S_m\}$ de conjuntos e um inteiro c , e o objetivo é decidir se existe uma subfamília $\mathcal{C} \subseteq \mathcal{S}$ desses conjuntos tal que $|\mathcal{C}| \leq c$ e $\cup_{S_j \in \mathcal{C}} S_j = U$, onde $U = S_1 \cup \dots \cup S_m$ (o conjunto \mathcal{C} é chamado de cobertura de U). Dada uma instância (\mathcal{S}, c) do problema de cobertura de conjuntos, vamos construir um grafo G e um inteiro K tal que existe uma cobertura $\mathcal{C} \subseteq \mathcal{S}$ de U com tamanho no máximo c se e somente se $g_d^s(G) \leq K$.

Definição 5.1 *Sejam inteiros $s, d \geq 0$. Definimos $p = p(s, d)$ e $q = q(s, d)$ como segue.*

$$p(s, d) = \begin{cases} d + \lceil \frac{d+1}{s-1} \rceil, & \text{se } s \leq d+1, \\ d+1, & \text{caso contrário.} \end{cases}$$

$$q(s, d) = \begin{cases} d + \lceil \frac{d}{s-1} \rceil, & \text{se } s \leq d+1, \\ 0, & \text{se } d+2 \leq s < 2d+2, \\ d, & \text{caso contrário.} \end{cases}$$

Seja (\mathcal{S}, c) uma instância do problema de cobertura de conjuntos, onde $\mathcal{S} = \{S_1, \dots, S_m\}$, e seja $U = \cup_{S_j \in \mathcal{S}} S_j = \{u_1, \dots, u_n\}$. Definimos $K = K_{s,d}(\mathcal{S}, c) = c$, se $d+1 \leq s \leq 2d+2$, e $K = K_{s,d}(\mathcal{S}, c) = c+1$, caso contrário. Seja $G = G_{s,d}(\mathcal{S}, c)$ o grafo definido como segue: para cada conjunto $S_j \in \mathcal{S}$, crie um vértice x_j em G e, para cada elemento $u_i \in U$, crie um caminho Y_i com p vértices $y_{i,1}, \dots, y_{i,p}$. Faça $\{x_1, \dots, x_m\}$ ser uma clique. Se $u_i \in S_j$, crie a aresta $x_j y_{i,1}$ em G . Crie um vértice z_0 e o ligue a todos os vértices x_1, \dots, x_m . Finalmente crie um caminho Z com q vértices z_1, \dots, z_q , e crie a aresta $z_0 z_1$, se $q > 0$.

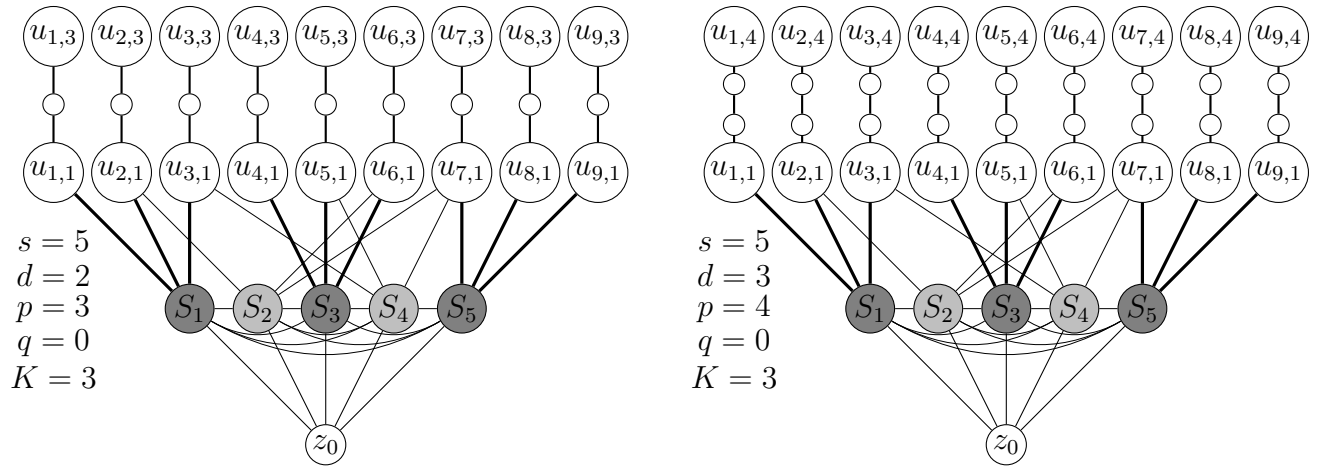
Veja as Figuras 24-26 para exemplos das reduções para cada caso.

Lema 5.1 *Dado um grafo G e um inteiro K , decidir se $g_d^s(G) \leq K$ é NP-difícil para todos inteiros $s > d > 0$ tais que $d+1 < s < 2d+2$.*

Prova: Como dito acima, faremos uma redução do problema de cobertura de conjuntos. Seja (\mathcal{S}, c) uma instância do problema de cobertura de conjuntos e seja $U = \bigcup_{S_j \in \mathcal{S}} S_j$. Da Definição 5.1, sejam $p = p(s, d) = d+1$, $q = q(s, d) = 0$, $G = G_{s,d}(\mathcal{S}, c)$ e $K = K_{s,d}(\mathcal{S}, c) = c$.

Primeiramente, suponha que não existe uma cobertura \mathcal{C} com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . A estratégia do espião para c policiais é se posicionar inicialmente em z_0 , correr para algum vértice em $\{y_{1,p}, \dots, y_{n,p}\}$, se comunicar

Figura 24 – Redução da instância (\mathcal{S}, c) do problema de cobertura de conjuntos, onde $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ e $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Casos para velocidade $s = 5$ e distância $d = 2, 3$. Ilustração da prova do Lema 5.1



Fonte: Elaborado pelo autor.

e depois retornar para z_0 , recomeçando o processo. Como são ao todo $K = c$ policiais e não existe cobertura de U com c conjuntos de \mathcal{S} , então existe algum i tal que não existe policial em $N[Y_i]$. Portanto, o espião vai para $y_{i,p}$ em um passo e consegue transmitir.

Agora, suponha que existe uma cobertura $\mathcal{C} = \{S_{j_1}, \dots, S_{j_c}\}$ com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . A estratégia dos policiais é descrita a seguir. Posicione inicialmente os c policiais nos vértices x_{j_1}, \dots, x_{j_c} . Como \mathcal{C} é uma cobertura, podemos definir para cada elemento u_i de U um valor $c(i)$ tal que $u_i \in S_{c(i)} \in \mathcal{C}$.

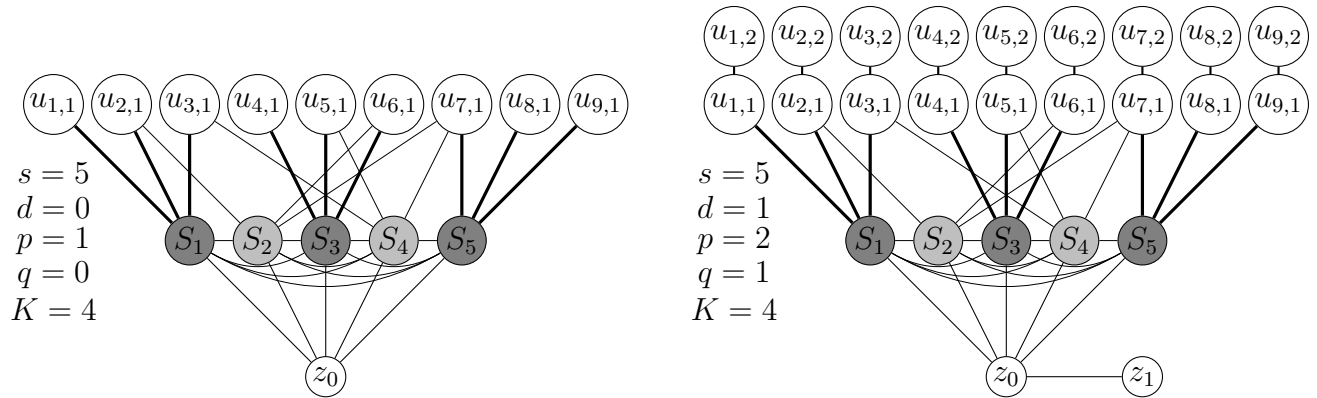
Se o espião não estiver em $\{y_{1,p}, \dots, y_{n,p}\}$, os policiais mantem as posições iniciais e portanto o espião não consegue transmitir. Se o espião estiver em algum vértice $y_{i,p}$, então o policial em $x_{c(i)}$ vai para $y_{i,1}$ (mantendo distância d do espião). Como $s < 2d + 2$, o espião não consegue ir de $y_{i,p}$ para um outro vértice $y_{j,p}$ em um passo ($j \neq i$). Portanto, se o espião sai de $y_{i,p}$, os policiais retomam as posições iniciais. Esta estratégia dos policiais evita que o espião consiga transmitir. \square

Lema 5.2 Dado um grafo G e um inteiro K , decidir se $g_d^s(G) \leq K$ é NP-difícil para todos inteiros $s > d \geq 0$ tais que $s \geq 2d + 2$.

Prova: Redução do problema de cobertura de conjuntos. Seja (\mathcal{S}, c) uma instância de cobertura de conjuntos e seja $U = \cup_{S_j \in \mathcal{S}} S_j$. Da Definição 5.1, sejam $p = p(s, d) = d + 1$, $q = q(s, d) = d$, $G = G_{s,d}(\mathcal{S}, c)$ e $K = K_{s,d}(\mathcal{S}, c) = c + 1$.

Primeiramente, suponha que não existe uma cobertura \mathcal{C} com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . A estratégia do espião para $c + 1$ policiais é se posicionar inicialmente em z_q , correr para algum vértice em $\{y_{1,p}, \dots, y_{n,p}\}$, se comunicar e depois retornar para z_q , recomeçando o processo. Considere que o espião está em z_q . Se não existe policial em $\{z_0, \dots, z_q\}$, então o espião consegue transmitir. Então suponha

Figura 25 – Redução da instância (\mathcal{S}, c) do problema de cobertura de conjuntos, onde, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ e $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Casos para velocidade $s = 5$ e distância $d \in \{0, 1\}$. Ilustrações da prova do Lemma 5.2



Fonte: Elaborado pelo autor.

que há um policial em z_0 . Como são ao todo $c + 1$ policiais, existem no máximo c policiais fora de $\{z_0, \dots, z_q\}$. Como não existe cobertura de U com c conjuntos de \mathcal{S} , então existe algum i tal que não existe policial em $N[Y_i]$. Portanto, o espião vai para $y_{i,p}$ em um passo e consegue transmitir.

Agora, suponha que existe uma cobertura $\mathcal{C} = \{S_{j_1}, \dots, S_{j_c}\}$ com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . A estratégia dos policiais é descrita a seguir. Posicione inicialmente os $c + 1$ policiais nos vértices $z_0, x_{j_1}, \dots, x_{j_c}$. Como \mathcal{C} é uma cobertura, podemos definir para cada elemento u_i de U um valor $c(i)$ tal que $u_i \in S_{c(i)} \in \mathcal{C}$.

Os únicos vértices de G que estão a distância maior que d de z_0 são os vértices $y_{1,p}, \dots, y_{n,p}$. Se o espião não estiver em $\{y_{1,p}, \dots, y_{n,p}\}$, os policiais mantem as posições iniciais e o espião não consegue transmitir. Se o espião estiver em $y_{i,p}$, então o policial em $x_{c(i)}$ vai para $y_{i,1}$ (mantendo distância d do espião) e o policial em z_0 vai para $x_{c(i)}$. Se o espião sai de $y_{i,p}$ para algum vértice $y_{j,p}$ tal que $c(i) = c(j)$, então o policial em $x_{c(i)}$ vai para $y_{j,1}$ (mantendo distância d do espião) e o policial em $y_{i,1}$ vai para $x_{c(i)}$. Se o espião sai de $y_{i,p}$ para algum vértice $y_{j,p}$ tal que $c(i) \neq c(j)$, então o policial em $x_{c(j)}$ vai para $y_{j,1}$ (mantendo distância d do espião), o policial em $x_{c(i)}$ vai para $x_{c(j)}$ e o policial em $y_{i,1}$ vai para $x_{c(i)}$. Se o espião sai de $y_{i,p}$ para algum vértice fora de $\{y_{1,p}, \dots, y_{n,p}\}$, então o policial em $x_{c(i)}$ vai para z_0 e o policial em $y_{i,1}$ vai para $x_{c(i)}$. Esta estratégia dos policiais evita que o espião consiga transmitir. \square

Vamos agora considerar o caso em que $d + 1 \geq s \geq 2$. O lema auxiliar abaixo será bastante utilizado.

Lema 5.3 *Dados s, d tais que $d + 1 \geq s \geq 2$, sejam $p = p(s, d) = d + \lceil \frac{d+1}{s-1} \rceil$ e $q = q(s, d) = d + \lceil \frac{d}{s-1} \rceil$. Seja $P_{s,d}$ o caminho formado por dois caminhos y_0, y_1, \dots, y_p e z_0, z_1, \dots, z_q com inclusão da aresta $y_0 z_0$. Suponha que o espião esteja em z_0 . Então, um policial*

consegue manter distância d do espião se o policial

- (a) estiver em z_0 e o espião se deslocar e parar em z_q ; ou
- (b) estiver em y_0 e o espião se deslocar e parar em y_p .

Além disso, o espião consegue distância maior que d do policial se o policial

- (c) estiver em y_0 e o espião se deslocar e parar em z_q ; ou
- (d) estiver em z_0 e o espião se deslocar e parar em y_p .

Logo, $g_d^s(P_{s,d}) = 2$. Finalmente, considere que um policial está em z_0 e o espião se desloca até parar em z_q . Seja $r = d \% (s - 1)$ o resto da divisão de d por $s - 1$. Então:

- (e) o policial consegue manter distância d do espião se o espião começar em z_r ; mas
- (f) o espião consegue distância maior que d do policial se começar em z_{r+1} (caso $r \geq 1$).

Prova: Considere inicialmente que o espião está em z_0 e começa a se deslocar para z_q . Suponha que um policial também está em z_0 . Quando o policial atinge o vértice z_j , o espião está no máximo na posição $z_{s \cdot j}$ e a distância entre eles é no máximo $j \cdot (s - 1)$. Portanto, se a distância entre eles é exatamente d , então o policial está no mínimo no vértice $z_{\lceil d/(s-1) \rceil}$ e, conseqüentemente, o espião está no mínimo no vértice $z_{d + \lceil d/(s-1) \rceil} = z_q$. Com isso, o policial consegue manter a distância d . No entanto, pelo exposto, se o policial começar em y_0 , o espião consegue se distanciar mais que d ao chegar em z_q . Portanto, se o espião estiver em z_0 , deve haver um policial em z_0 .

Considere agora que o espião está em z_0 e começa a se deslocar para y_p . Suponha que um policial está em y_0 . Quando o policial atinge o vértice y_j , o espião está no máximo na posição $y_{s \cdot j - 1}$ e a distância entre eles é no máximo $j \cdot (s - 1) - 1$. Portanto, se a distância entre eles é exatamente d , então o policial está no mínimo no vértice $y_{\lceil (d+1)/(s-1) \rceil}$ e conseqüentemente o espião está no mínimo no vértice $y_{d + \lceil (d+1)/(s-1) \rceil} = y_p$. Com isso, o policial consegue manter a distância d . No entanto, pelo exposto, se o policial começar em z_0 , o espião consegue se distanciar mais que d ao chegar em y_p . Portanto, se o espião estiver em z_0 , deve haver um policial em y_0 .

Com isso, dois policiais são necessários e suficientes. Logo, $g_d^s(P_{s,d}) = 2$.

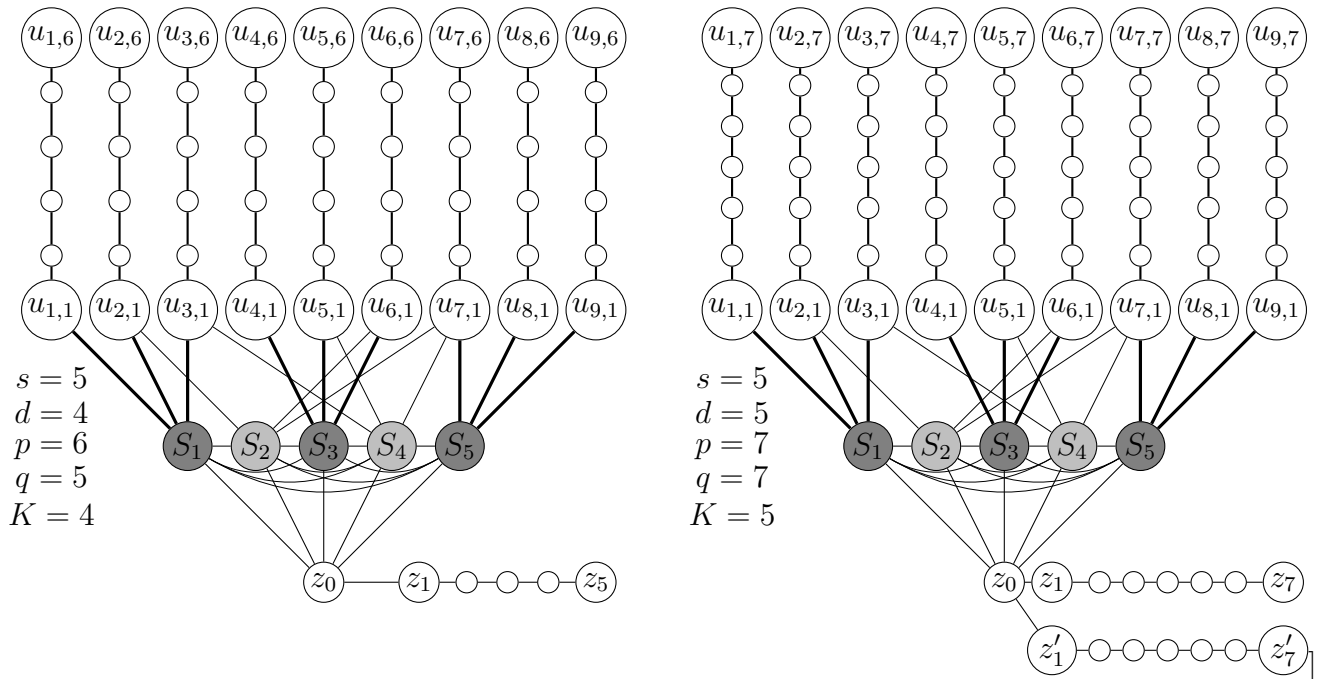
Finalmente, suponha que um policial esteja em z_0 e o espião esteja em z_r , onde $r = d \% (s - 1)$ é o resto da divisão de d por $s - 1$, e começa a se deslocar para z_q . O caso $r = 0$, já foi tratado antes. Então assumamos que $r \geq 1$. Quando o policial atinge o vértice z_j , o espião está no máximo na posição $z_{s \cdot j + r}$ e a distância entre eles é no máximo $j \cdot (s - 1) + r$. Portanto, se a distância entre eles é exatamente d , então o policial está no mínimo no vértice $z_{\lceil (d-r)/(s-1) \rceil} = z_{\lceil d/(s-1) \rceil - 1}$ e conseqüentemente o espião está no mínimo no vértice $z_{d + \lceil d/(s-1) \rceil - 1} = z_{q-1}$. Como o espião termina em z_q , o policial consegue manter a distância d .

Se o espião começar em z_{r+1} e $r \geq 1$, então, se a distância entre eles é exatamente $d - 1$, o policial consegue atingir o vértice $z_{\lceil (d-1-(r+1))/(s-1) \rceil} = z_{\lceil d/(s-1) \rceil - 1}$ e conseqüentemente o espião atinge o vértice $z_{d-1 + \lceil d/(s-1) \rceil - 1} = z_{q-2}$. Como o espião termina em z_q , a distância passa de $d - 1$ para $d + 1$ e o espião consegue se comunicar.

□

Agora vamos considerar o caso em que $s \leq d+1$ e $r = d \bmod (s-1) \geq \lceil \frac{s}{2} \rceil - 1$ ou $r = 0$.

Figura 26 – Redução de uma instância (\mathcal{S}, c) do problema de cobertura de conjuntos, onde $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ e $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Casos para velocidade $s = 5$ e distância $d \in \{4, 5\}$. Ilustração das provas do Lema 5.4 (esquerda) e Lema 5.5 (direita)



Fonte: Elaborado pelo autor.

Lema 5.4 Dado um grafo G e um inteiro K , decidir se $g_d^s(G) \leq K$ é NP-difícil para todos inteiros $s, d > 0$ tais que $s \leq d+1$ e $d \% (s-1) \in \{\lceil \frac{s}{2} \rceil - 1, \dots, s-2, 0\}$, onde $d \% (s-1)$ é o resto da divisão de d por $s-1$.

Prova: Redução do problema de cobertura de conjuntos. Seja (\mathcal{S}, c) uma instância de cobertura de conjuntos e seja $U = \cup_{S_j \in \mathcal{S}} S_j$. Da Definição 5.1, sejam $p = p(s, d) = d + \lceil \frac{d+1}{s-1} \rceil$, $q = q(s, d) = d + \lceil \frac{d}{s-1} \rceil$, $G = G_{s,d}(\mathcal{S}, c)$ e $K = K_{s,d}(\mathcal{S}, c) = c + 1$.

Primeiramente, suponha que não existe uma cobertura \mathcal{C} com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . A estratégia do espião para $c+1$ policiais é se posicionar inicialmente em z_0 , correr para algum vértice em $\{z_q, y_{1,p}, \dots, y_{n,p}\}$, se comunicar e depois retornar para z_0 , recomeçando o processo. Considere que o espião está em z_0 . Se não existe policial em $\{z_0, \dots, z_q\}$, então, pelo Lema 5.3(c), o espião consegue transmitir se deslocando até z_q . Então suponha que há um policial em z_0 . Como são ao todo $c+1$ policiais, existem no máximo c policiais fora de $\{z_0, \dots, z_q\}$. Como não existe cobertura de U com c conjuntos de \mathcal{S} , então existe algum i tal que não existe policial em $N[Y_i]$. Portanto, pelo Lema 5.3(d), o espião consegue transmitir se deslocando até $y_{i,p}$.

Agora, suponha que existe uma cobertura $\mathcal{C} = \{S_{j_1}, \dots, S_{j_c}\}$ com c conjuntos

em \mathcal{S} que cobrem todos os elementos de U . Posicione inicialmente os $c + 1$ policiais nos vértices $z_0, x_{j_1}, \dots, x_{j_k}$. Como \mathcal{C} é uma cobertura, podemos definir para cada elemento u_i de U um valor $c(i)$ tal que $u_i \in S_{c(i)} \in \mathcal{C}$. Seja ainda $r = d \% (s - 1)$ o resto da divisão de d por $s - 1$. A estratégia dos policiais consiste em sempre ter um policial em cada x_{j_1}, \dots, x_{j_k} .

Suponha que o espião esteja em um vértice de $\{z_0, x_1, \dots, x_m\}$. Então, os policiais devem estar nas posições iniciais: $z_0, x_{j_1}, \dots, x_{j_k}$. Caso o espião se desloque para o vértice z_q , o policial em z_0 consegue manter distância d do espião pelo Lema 5.3(a). Caso o espião se desloque para algum vértice $y_{i,p}$, o policial em $x_{c(i)}$ o persegue (mantendo pelo Lema 5.3(b) distância d do espião) e o policial em z_0 vai para $x_{c(i)}$. Consequentemente, se o espião estiver em algum vértice $y_{i,\ell}$ e se deslocar para z_q , o policial em z_0 consegue manter distância d do espião e, se o espião estiver em algum vértice z_ℓ e se deslocar para $y_{i,p}$, o policial em $x_{c(i)}$ consegue manter distância d do espião.

Falta então analisar o caso em que o espião esteja em um vértice $y_{i,\ell}$, não há policial em z_0 e o espião se desloca para z_q . Se $d \% (s - 1) = 0$ (ou seja, d é múltiplo de $s - 1$), então $p = q + 1$ e consequentemente o policial em $x_{c(i)}$ consegue manter distância d do espião pelo Lema 5.3(b). Portanto, assumamos que $d \% (s - 1) \geq \frac{s}{2} - 1$. Consequentemente $p = q = d + \lceil d/(s - 1) \rceil$.

Pelos itens (e) e (f) do Lema 5.3, temos que o policial em $x_{c(i)}$ só precisa entrar em perseguição do espião em $y_{i,\ell}$ se $\ell > r$. Se $\ell \leq r$, os policiais podem manter suas posições iniciais (e portanto haverá um policial em z_0). Então assumamos que $\ell = r + 1 \geq \frac{s}{2}$.

Neste caso, em um passo, o espião consegue atingir no máximo o vértice $z_{s-\ell-1}$. Como $s - \ell - 1 \leq s/2 - 1 \leq r$, então o policial em $x_{c(i)}$ vai para z_0 e consegue manter distância d do espião pelo Lema 5.3(e). \square

Lema 5.5 *Dado um grafo G e um inteiro K , decidir se $g_d^s(G) \leq K$ é NP-difícil para todos inteiros $s, d > 0$ tais que $s \leq d + 1$ e $1 \leq d \% (s - 1) < \frac{s}{2} - 1$, onde $d \% (s - 1)$ é o resto da divisão de d por $s - 1$.*

Prova: Redução do problema de cobertura de conjuntos. Seja (\mathcal{S}, c) uma instância de cobertura de conjuntos e seja $U = \cup_{S_j \in \mathcal{S}} S_j$. Da Definição 5.1, sejam $p = p(s, d) = d + \lceil \frac{d+1}{s-1} \rceil$, $q = q(s, d) = d + \lceil \frac{d}{s-1} \rceil$ e $G = G_{s,d}(\mathcal{S}, c)$. Acrescente a G mais um caminho z'_1, \dots, z'_q e ligue a aresta $z_0 z'_1$. Seja $K = c + 2$. Note que, como $d \% (s - 1) \neq 0$, então $p = q$. Seja ainda $r = d \% (s - 1)$.

Primeiramente, suponhamos que não existe uma cobertura \mathcal{C} com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . A estratégia do espião para $c + 2$ policiais é se posicionar inicialmente em z'_{r+1} , correr para algum vértice em $\{z_q, z'_q, y_{1,p}, \dots, y_{n,p}\}$, se comunicar e depois retornar para z'_{r+1} , recomeçando o processo. Considere que o espião está em z'_{r+1} . Se não existe policial em $\{z'_1, \dots, z'_q\}$, então, pelo Lema 5.3(f), o espião consegue transmitir se deslocando até z'_q . Além disso, se não existe um policial em $\{z_0, \dots, z_q\}$, o espião consegue em um passo atingir z_{r+1} e, pelo Lema 5.3(f), conseguiria transmitir se

deslocando até z_q . Então suponha que há um policial em z_0 e outro em z'_1 . Como são ao todo $c + 2$ policiais, existem no máximo c policiais fora de $\{z_0, z_1, z'_1, \dots, z_q, z'_q\}$. Como não existe cobertura de U com c conjuntos de \mathcal{S} , então existe algum i tal que não existe policial em $N[Y_i]$. Portanto, o espião consegue em um passo atingir $y_{i,r+1}$ e, pelo Lema 5.3(f), o espião consegue transmitir se deslocando até $y_{i,p}$.

Agora, suponha que existe uma cobertura $\mathcal{C} = \{S_{j_1}, \dots, S_{j_c}\}$ com c conjuntos em \mathcal{S} que cobrem todos os elementos de U . Posicione inicialmente os $c + 2$ policiais nos vértices $z_0, z_0, x_{j_1}, \dots, x_{j_k}$ (dois policiais em z_0). Como \mathcal{C} é uma cobertura, podemos definir para cada elemento u_i de U um valor $c(i)$ tal que $u_i \in S_{c(i)} \in \mathcal{C}$. A estratégia dos policiais consiste em sempre ter um policial em cada $z_0, x_{j_1}, \dots, x_{j_k}$.

Suponha que o espião esteja em um vértice de $\{z_0, x_1, \dots, x_m\}$. Então, os policiais devem estar nas posições iniciais: $z_0, z_0, x_{j_1}, \dots, x_{j_k}$. Caso o espião se desloque para o vértice z_q , um policial em z_0 o persegue, mantendo distância d do espião pelo Lema 5.3(a)). O mesmo vale para z'_q . Caso o espião se desloque para algum vértice $y_{i,p}$, o policial em $x_{c(i)}$ o persegue (mantendo pelo Lema 5.3(b) distância d do espião) e um policial em z_0 vai para $x_{c(i)}$. Conseqüentemente, se o espião estiver em algum vértice $y_{i,\ell}$ e se deslocar para z_q , um policial em z_0 consegue manter distância d do espião e, se o espião estiver em algum vértice z_ℓ ou z'_ℓ e se deslocar para $y_{i,p}$, o policial em $x_{c(i)}$ consegue manter distância d do espião.

Suponha então que o espião esteja em $\{y_{i,1}, \dots, y_{i,p}\}$ e, em um passo, entre em $\{y_{i',1}, \dots, y_{i',p}\}$. Se $c(i) = c(i')$, então o policial em $x_{c(i)}$ vai para $y_{i',1}$ e o policial em $y_{i',1}$ vai para $x_{c(i)}$. Caso contrário, o policial em $x_{c(i')}$ vai para $y_{i',1}$, o policial em $x_{c(i)}$ vai para $x_{c(i')}$ e o policial em $y_{i,1}$ vai para $x_{c(i)}$.

Finalmente suponha que o espião esteja em $\{y_{i,1}, \dots, y_{i,p}\}$ e, em um passo, entre em $\{z_1, \dots, z_q\}$. Então, o policial em z_0 vai para z_1 , o policial em $x_{c(i)}$ vai para z_0 e o policial em $y_{i,1}$ vai para $x_{c(i)}$. Simetricamente, o mesmo vale para $\{z'_1, \dots, z'_q\}$. \square

Feige provou que o problema de cobertura de conjuntos é $O(\log n)$ -inaproximável em tempo polinomial, a menos que $P = NP$ (FEIGE, 1998). Portanto os Lemas 5.1, 5.2, 5.4 e 5.5 implicam o seguinte Teorema.

Teorema 5.1 *Sejam $s \geq 2$ e $d \geq 0$ dois inteiros fixos. Dado um grafo G e um inteiro $k \in \mathbb{N}$ decidir se $g_d^s(G) \leq k$ é NP-difícil e $O(\log n)$ -inaproximável em tempo polinomial, a menos que $P = NP$.*

5.3 Complexidade para o caso direcionado

Nesta seção consideramos uma variante do jogo do espião em grafos direcionados. Nessa variante, tanto os guardas como o espião devem se mover seguindo a orientação dos arcos. No entanto, a distância d é medida no grafo desconsiderando as direções das arestas. Podemos imaginar que os arcos possuem restrições de movimentação (orientação

de uma rua), mas tais restrições são irrelevantes em relação a visão que os guardas possuem do espião. Isto é, embora os guardas não possam se movimentar de maneira contrária a direção dos arcos, eles podem observar o que o espião está fazendo até uma distância no máximo d , o que o impede de se comunicar sem estragar seu disfarce.

Dado um digrafo D e inteiros $s > 0$ e $d \geq 0$, o (s, d) -guard-number de D , $\bar{g}_d^s(G)$, é o menor número de guardas necessários para vencer o Jogo do Espião direcionado em D , onde s é a velocidade do espião e d é a distância máxima que um guarda pode vigiar.

Mostramos que, dado um digrafo D e inteiros $k > 0$, $s \geq 1$ e $d = 2$, decidir se $\bar{g}_d^s(D) \leq k$ é PSPACE-difícil mesmo quando D é um DAG.

O resultado é obtido através de uma redução do famoso *Problema da Fórmula Quantificada na Forma Normal Conjuntiva (QBF)*. Dados um conjunto de variáveis booleanas x_1, \dots, x_n e uma fórmula booleana $F = C_1 \wedge C_2 \wedge \dots \wedge C_m$, onde todo C_j é uma disjunção de literais. O problema *QBF* consiste em decidir se a expressão $\phi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n F$, onde todo Q_i é um quantificador do tipo \forall ou \exists , é verdadeira. *QBF* é um famoso problema PSPACE-completo (STOCKMEYER and MEYER, 1973).

Teorema 5.2 *Dado um digrafo D e inteiros $k, s > 0$. O problema de decidir se $\bar{g}_2^s(D) \leq k$, na versão em que os guardas são posicionados antes do espião, é PSPACE-difícil, mesmo quando D é um DAG.*

Prova: Seja ϕ um fórmula booleana quantificada com n variáveis. Construímos um DAG D_ϕ tal que ϕ é verdade se, e somente se, n guardas podem controlar um espião com velocidade s a uma distância 2 em D_ϕ após uma quantidade finita de turnos.

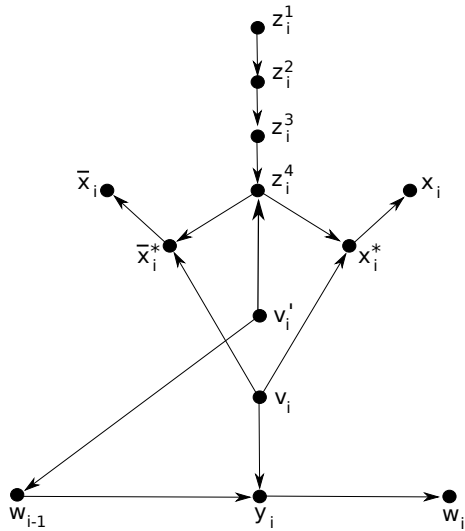
Para cada $Q_i x_i$ de ϕ construímos um gadget D_i . Se $Q_i = \exists$, então $V(D_i) = \{w_{i-1}, z_i^1, z_i^2, z_i^3, z_i^4, x_i, x_i^*, \bar{x}_i, \bar{x}_i^*, y_i, v_i, v_i', w_i\}$ e os arcos entre os vértices do gadget são mostrados na Figura 27. Se $Q_i = \forall$, então $V(D_i) = \{w_{i-1}, z_i^1, z_i^2, z_i^3, z_i^4, x_i, x_i^*, \bar{x}_i, \bar{x}_i^*, y_i, \bar{y}_i, v_i, \bar{v}_i, v_i', w_i\}$ os arcos entre os vértices do gadget são mostrados na Figura 28.

Observe que o vértice w_i aparece em D_i e em D_{i+1} . Falta-nos estabelecer uma relação entre cada cláusula e as literais que esta contém. Para cada cláusula C_i criamos o vértice c_i em D_ϕ e adicionamos um arco de w_n para c_i . Também adicionamos um arco de c_i para x_i (\bar{x}_i) se a cláusula C_i contém a literal x_i (\bar{x}_i).

Um exemplo de digrafo D_ϕ onde $\phi = \exists x_1 \forall x_2 (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2)$ é mostrado na Figura 29.

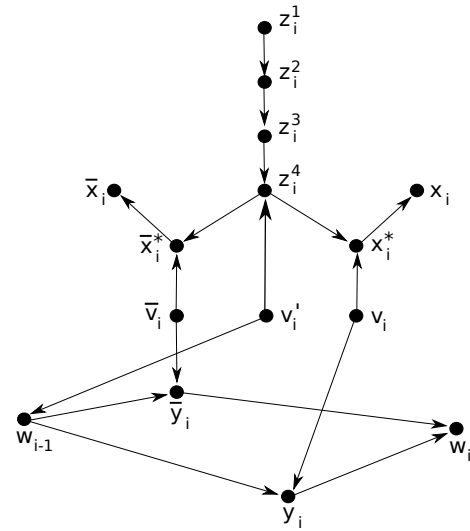
Resta-nos mostrar que ϕ é verdade se, e somente se, $\bar{g}_2^s(D_\phi) = n$. Primeiro note que, para cada gadget D_i , pelo menos um guarda deve escolher um vértice de $S_i = \{z_i^1, z_i^2, z_i^3\}$ como sua posição inicial. Caso contrário o espião escolheria z_i^1 como sua posição inicial e nenhum guarda poderia alcançar uma distância 2 de tal vértice e desta forma o espião iria vencer. Iremos nos referir ao guarda inicialmente em S_i como p_i . Uma vez que D_ϕ possui n gadgets D_i , então $\bar{g}_2^s(D_\phi) \geq n$. Ademais, assumindo que cada guarda p_i inicia o jogo em z_i^1 ele só pode ocupar os vértices do conjunto $R_i = \{z_i^1, z_i^2, z_i^3, z_i^4, x_i, x_i^*, \bar{x}_i, \bar{x}_i^*\}$ durante todo o resto da partida.

Figura 27 – $D_i(\exists)$



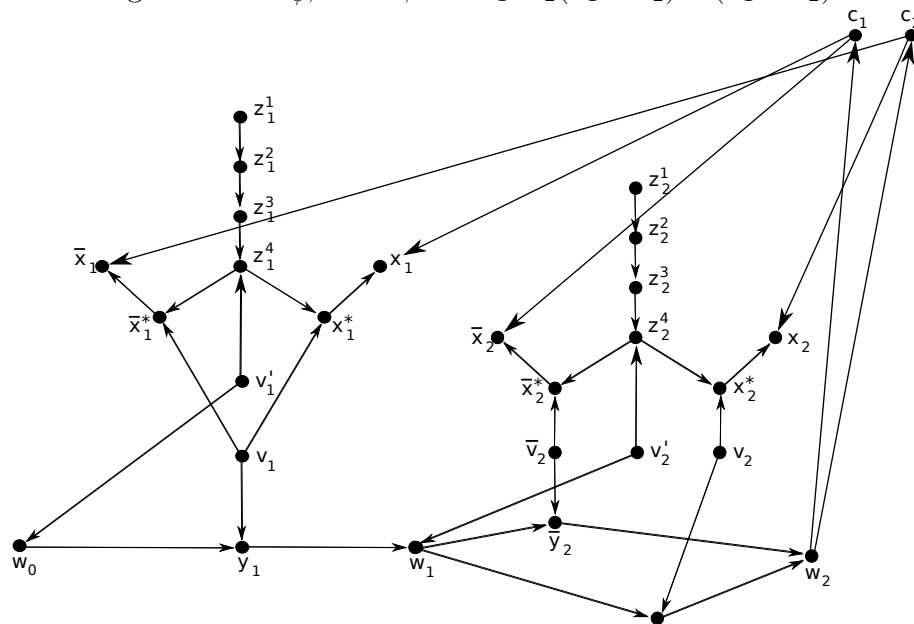
Fonte: Elaborado pelo autor.

Figura 28 – $D_i(\forall)$



Fonte: Elaborado pelo autor.

Figura 29 – D_ϕ , onde $\phi = \exists x_1 \forall x_2 (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2)$



Fonte: Elaborado pelo autor.

Suponha que $\phi = falso$. Iremos descrever uma estratégia vencedora para um espião jogando contra n guardas. Assuma que há exatamente um guarda em cada conjunto S_i , isto é, o espião não pode vencer apenas se posicionando inicialmente em um vértice z_i^1 desprotegido. O espião então se posiciona inicialmente no vértice w_0 . Agora, suponha que o espião está em um vértice w_{i-1} de $D_i(\forall)$, então o único guarda que pode alcançar um vértice a distância 2 de w_{i-1} é p_i quando este ocupar o vértice z_i^4 . O espião espera até que o guarda p_i se mova para z_i^4 , se o guarda nunca o fizer, então o espião permanece em w_{i-1} e vence o jogo. Desta maneira suponha que p_i eventualmente se desloque para z_i^4 , então o espião escolhe entre se mover para y_i ou \bar{y}_i , dependendo da

escolha do espião o guarda p_i é forçado então a se mover para x_i^* ou para \bar{x}_i^* , pois estes são os únicos vértices alcançáveis para qualquer guarda que estão à distância no máximo 2 de y_i e \bar{y}_i respectivamente. Se p_i se desloca para x_i^* a variável correspondente a x_i é considerada *verdade*. Caso contrário, se p_i se desloca para \bar{x}_i^* então $x_i = falso$. Isto significa que para uma variável quantificada $\forall x_i$ o espião escolhe o valor de x_i .

Se o espião está em um vértice w_{i-1} de um gadget $D_i(\exists)$, novamente, o único guarda que pode alcançar um vértice a distância máxima 2 do espião é p_i quando este ocupar o vértice z_i^4 . O espião então espera o guarda p_i se mover para z_i^4 e então se desloca para y_i , desta vez p_i não é forçado a se mover especificamente para x_i^* ou para \bar{x}_i^* , mas ainda assim deve escolher um dos dois. Novamente, se p_i se move para x_i^* , a variável correspondente x_i é considerada *verdade*, caso contrário, se p_i se move para \bar{x}_i^* então $x_i = falso$. Isto significa que para uma variável quantificada $\exists x_i$ os guardas escolhem o valor de x_i .

Quando p_n se move para x_n^* ou \bar{x}_n^* cada guarda está em um x_i^* (\bar{x}_i^*) ou x_i (\bar{x}_i). Observe que cada guarda só pode monitorar os vértices c_j correspondendo às cláusulas que contém o literal que ele tornou verdadeiro. Como $\phi = false$ então o espião pode escolher entre y_i e \bar{y}_i nos gadgets do tipo $D_i(\forall)$ de tal maneira que independente de como os guardas escolham x_i^* ou \bar{x}_i^* nos gadgets do tipo $D_i(\exists)$ há pelo menos um vértice c_j que não pode ser protegido por nenhum guarda. O espião então se move para este vértice, permanece nele e vence o jogo.

Suponha que $\phi = verdade$. Neste caso, devemos mostrar uma estratégia vencedora para os guardas. No primeiro turno cada guarda p_i , $i = 1, \dots, n$, escolhe z_i^3 como sua posição inicial. Se o espião escolhe como sua posição inicial $z_i^1, z_i^2, z_i^3, z_i^4, x_i^*$ ou \bar{x}_i^* o guarda p_i não precisa se mover uma vez que o espião está à distância no máximo 2 de z_i^3 . Os únicos vértices para os quais o espião pode ir a partir dessas posições iniciais que não estão sob a proteção de p_i são x_i ou \bar{x}_i . Se o espião for para qualquer um deles o guarda p_i apenas move-se para z_i^4 . Como o espião não pode se mover e está à distância no máximo 2 de um guarda, os guardas vencem o jogo.

Se o espião escolhe algum vértice v_i, \bar{v}_i ou v_i' como sua posição inicial, então p_i se desloca para z_i^4 , após isso, se o espião vai para x_i^*, \bar{x}_i^* ou z_i^4 , então p_i segue a mesma estratégia descrita acima e, desta forma, os guardas vencem. Portanto o espião, independente de sua posição inicial, deve eventualmente se deslocar para um vértice w_i, y_i, \bar{y}_i ou algum vértice de cláusula c_j , caso contrário ele é derrotado.

Suponha que o espião está em algum vértice w_{i-1} de $D_i(\forall)$, então o guarda p_i se move para z_i^4 e impede o espião de se comunicar. A única opção do espião é mover-se para y_i ou \bar{y}_i forçando p_i a mover-se para x_i^* ou \bar{x}_i^* respectivamente. Novamente, para uma variável quantificada $\forall x_i$ o espião escolhe o valor de x_i . Depois que o espião se deslocar de y_i (\bar{y}_i) para w_i o guarda p_i vai para o vértice x_i (\bar{x}_i) e permanece lá para sempre.

De maneira similar, se o espião está em algum vértice w_{i-1} de $D_i(\exists)$ então o

guarda p_i se movimenta para z_i^4 e impede o espião de se comunicar. O espião deve então se deslocar para y_i , neste caso p_i não é forçado a se mover especificamente para x_i^* ou \bar{x}_i^* , mas ainda deve escolher um dos dois como sua próxima posição. Portanto, para uma variável quantificada $\exists x_i$ o guarda p_i escolhe o valor de x_i . Após o espião se mover de y_i para w_i o guarda p_i se desloca para x_i ou \bar{x}_i dependendo de seu movimento anterior e permanece neste vértice para sempre.

Observe que após o espião se deslocar de y_n ou \bar{y}_n para w_n todo guarda está à distância 2 do ladrão, à distância 1 de todo vértice de cláusula que contém o literal que ele escolheu tornar verdadeiro e à distância 2 de cada um dos literais destas cláusulas. Como $\phi = \text{verdadeiro}$, então os guardas podem escolher entre y_i e \bar{y}_i nos gadgets do tipo $D_i(\exists)$ de tal forma que, não importa como o espião escolha x_i^* ou \bar{x}_i^* nos gadgets $D_i(\forall)$, todos os vértices de cláusula estão a distância 1 de pelo menos 1 guarda. Portanto os únicos vértices ainda alcançáveis para o espião estão à distância no máximo 2 dos guardas que, desta forma, vecem o jogo. \square

Destacamos que a prova acima é independente das velocidades do espião e dos guardas, uma vez que ambos não se beneficiariam com uma velocidade maior que 1.

5.4 Limites para caminhos e ciclos

Como comprovamos nas seções anteriores, o jogo do espião é um problema bastante difícil. Além disso, por se tratar de um problema recente, pouco se sabe sobre o problema mesmo quando o tratamos para grafos simples. Assim, nesta seção iremos estudar estratégias para os guardas e para o espião em dois tipos simples de grafos: caminhos e ciclos. Para facilitar a leitura, assim como na seção anterior, as provas irão se restringir ao casos em que a velocidade do espião é igual a 2. No entanto, provas para limites de $g_d^s(C_n)$ e o valor exato de $g_d^s(P_n)$ para $s \geq 2$ podem ser encontrados em (COHEN *et al.*, 2018).

O teorema a seguir é consequência direta dos dois lemas seguintes.

Teorema 5.3 *Para qualquer caminho P com $n + 1$ vértices e para quaisquer inteiros $k \geq 1$ e $s \geq 2$,*

$$\left\lfloor \frac{n(s-1)}{2ks} \right\rfloor \leq d_k^s(P_n) \leq \left\lceil \frac{(n+1)(s-1)}{2ks} \right\rceil$$

Lema 5.6 *Para qualquer caminho P com $n + 1$ vértices e para quaisquer inteiros $k \geq 1$ e $s \geq 2$, $d_k^s(P) \geq \lfloor \frac{n(s-1)}{2ks} \rfloor$.*

Prova: Para facilitar a leitura, provamos o lema no caso em que $\frac{2d-1}{s-1} \in \mathbb{N}$.

Sejam $P = (v_0, v_1, \dots, v_n)$ e $d = \lfloor \frac{n(s-1)}{2ks} \rfloor$. Mostramos que um espião com velocidade s jogando contra no máximo k guardas pode alcançar um vértice a distância pelo menos d de qualquer guarda. Em resumo, a estratégia do espião consiste em, começando em uma das extremidades de P , correr a toda velocidade em direção à outra extremidade. Mostraremos que, seguindo essa estratégia, haverá um turno em que o espião

estará a distância pelo menos d de todos os guardas e, portanto, $d_{s,k}(P) \geq d$.

De maneira mais formal, a estratégia dos espião é a seguinte. Inicialmente o espião está ocupando uma das extremidades do caminho, por exemplo v_0 . Então, no turno $i \geq 1$, o espião se move de $v_{i(s-1)}$ para v_{is} .

Iremos mostrar por indução em $1 \leq i \leq k$, que após o turno $i \frac{2d-1}{s-1}$ (quando o espião ocupará $v_{si \frac{2d-1}{s-1}}$), ou pelo menos i guardas estão ocupando vértices em $\{v_0, \dots, v_{si \frac{2d-1}{s-1} - d}\}$, ou houve um turno $0 \leq j < i \frac{2d-1}{s-1}$ tal que, após a movimentação dos guardas neste turno, a distância entre o espião e todos os guardas era pelo menos d .

Inicialmente, deve haver pelo menos um guarda, digamos g_1 , ocupando um vértice em $\{v_0, \dots, v_{d-1}\}$ pois, caso contrário, todos os guardas estão a distância pelo menos d no turno 0. Portanto, após o turno $\frac{2d-1}{s-1}$, o guarda g_1 estará ocupando um vértice em $\{v_0, \dots, v_{\frac{2d-1}{s-1} + d - 1}\} = \{v_0, \dots, v_{s \frac{2d-1}{s-1} - d}\}$ enquanto o espião estará ocupando $v_{s \frac{2d-1}{s-1}}$. Assim, a hipótese indutiva é válida para $i = 1$. Note que o espião está a distância pelo menos d de g_1 .

Seja $1 \leq i \leq k$ e assuma por indução que, após o turno $i \frac{2d-1}{s-1}$, há pelo menos i guardas ocupando os vértices em $\{v_0, \dots, v_{si \frac{2d-1}{s-1} - d}\}$. Ademais, pela definição da estratégia do espião, ele estará ocupando o vértice $v_{si \frac{2d-1}{s-1}}$. Observe que todos os i guardas estão à distância pelo menos d do espião.

Então, após o turno $i \frac{2d-1}{s-1}$, deve haver pelo menos um guarda, digamos g_{i+1} , ocupando um vértice em $\{v_{si \frac{2d-1}{s-1} - d + 1}, \dots, v_{si \frac{2d-1}{s-1} + d - 1}\}$, pois, caso contrário, todos os guardas estariam a distância pelo menos d do espião no turno i . Portanto, após o turno $(i+1) \frac{2d-1}{s-1}$, o guarda g_{i+1} estará ocupando um vértice em $\{v_0, \dots, v_{(s(i+1)) \frac{2d-1}{s-1} + d - 1}\}$, isto é $\{v_0, \dots, v_{s(i+1) \frac{2d-1}{s-1} - d}\}$, e o espião estará ocupando o vértice $v_{(i+1)s \frac{2d-1}{s-1}}$. Ademais, todos os i guardas que estavam ocupando vértices em $\{v_0, \dots, v_{si \frac{2d-1}{s-1}}\}$ após o turno $i \frac{2d-1}{s-1}$ estarão ocupando vértices em $\{v_0, \dots, v_{s(i+1) \frac{2d-1}{s-1} - d}\}$ após o turno $(i+1) \frac{2d-1}{s-1}$. Assim, a hipótese indutiva é válida para $i+1$.

Portanto, após o turno $k \frac{2d-1}{s-1}$, houve um turno anterior quando o espião estava à distância pelo menos d de todos os guardas, ou todos os k guardas estão ocupando vértices em $\{v_0, \dots, v_{sk \frac{2d-1}{s-1} - d}\}$ enquanto o espião ocupa o vértice $v_{ks \frac{2d-1}{s-1}}$ (observe que tal vértice existe uma vez que $ks \frac{2d-1}{s-1} \leq n$ pela definição de d), neste caso o espião está distância pelo menos d de todos os guardas neste turno. \square

Lema 5.7 *Para todo caminho P com $n+1$ e quaisquer $k \geq 1$, $s \geq 2$,*

$$d_{s,k}(P) \leq \left\lceil \frac{(n+1)(s-1)}{2ks} \right\rceil.$$

Prova: Para facilitar a leitura, provamos o Lema no caso em que $s = 2$.

É suficiente provar o resultado para o caso em que $d = \frac{n+1}{4k} \in \mathbb{N}$. Seja $P = (v_0, \dots, v_n)$ e, para todo $1 \leq i \leq k$, seja $P_i = (v_{4(i-1)d}, \dots, v_{4di})$.

Nós mostraremos uma estratégia que garante que k guardas podem manter um

espião com velocidade s sempre à distância no máximo d de pelo menos um dos guardas. O i -ésimo guarda é designado para o subcaminho P_i (este guarda se move apenas nos vértices de P_i). Ademais, um guarda i só irá se mover em um determinado turno se o espião se movimentou por uma aresta de P_i no turno em questão (observe que os subcaminhos P_i não são disjuntos em arestas).

Seja $i \leq k$ tal que o espião ocupa o vértice $x = v_{(4i-2)d+\ell}$ com $-2d \leq \ell \leq 2d$. Isto é, $x \in P_i$. Assuma que:

- Para qualquer $1 \leq j < i$, o j -ésimo guarda ocupa $v_{(4j-1)d}$;
- Para qualquer $i < j \leq k$, o j -ésimo guarda ocupa $v_{(4j-3)d}$;
- O i -ésimo guarda ocupa $v_{(4i-2)d+\lceil \ell/2 \rceil}$ se $\ell \geq 0$ e $v_{(4i-2)d+\lfloor \ell/2 \rfloor}$ se $\ell \leq 0$.

Certamente, se estas condições são satisfeitas, o espião está à distância no máximo $\lceil |\ell|/2 \rceil \leq d$ do i -ésimo guarda. Ademais, tais posições podem ser escolhidas pelos guardas uma vez que o espião escolheu sua posição inicial.

A seguir mostramos que, qualquer que seja o movimento do espião, os guardas podem manter esta configuração de posições. Seja y o próximo vértice a ser ocupado pelo espião. Note que $y = v_{(4i-2)d+\ell+a}$ onde $a \in \{-2, -1, 0, +1, +2\}$.

Tratamos primeiro o caso em que x e y não estão no mesmo subcaminho P_i . Isso pode acontecer em apenas dois casos: ou $x = v_{4id-1}$ e $y = v_{4id+1}$ ($\ell = 2d-1$ e $a = +2$) ou $x = v_{4(i-1)d+1}$ e $y = v_{4(i-1)d-1}$ ($\ell = -2d+1$ e $a = -2$). No primeiro caso, o i -ésimo guarda vai de $v_{(4i-1)d-1}$ para $v_{(4i-1)d}$ e o $(i+1)$ -ésimo guarda vai de $v_{(4(i+1)-3)d} = v_{(4i+1)d}$ para $v_{(4i+1)d+1}$. No último, o i -ésimo guarda vai de $v_{(4i-3)d+1}$ para $v_{(4i-3)d}$ e o $(i-1)$ -ésimo guarda vai de $v_{(4(i-1)-1)d}$ para $v_{(4(i-1)-1)d-1}$. Em ambos os casos as condições se mantêm válidas.

A partir de agora, admita que x e y pertencem a P_i . Neste caso, apenas o i -ésimo pode se mover. Diversos casos são possíveis dependendo dos valores de $a \in \{-2, -1, 0, +1, +2\}$ e ℓ .

- Se $\ell \geq 0$ e $\ell + a \geq 0$, então

$$v_{(4i-2)d+\lfloor (\ell+a)/2 \rfloor} \in \{v_{(4i-2)d+\lfloor \ell/2 \rfloor-1}, v_{(4i-2)d+\lfloor \ell/2 \rfloor}; v_{(4i-2)d+\lfloor \ell/2 \rfloor+1}\}.$$

Assim, qualquer que seja o movimento do espião, o i -ésimo guarda pode ir de $v_{(4i-2)d+\lfloor \ell/2 \rfloor}$ para $v_{(4i-2)d+\lfloor (\ell+a)/2 \rfloor}$ ou se movendo para um de seus vizinhos, ou permanecendo estacionário.

- Se $\ell \leq 0$ e $\ell + a \leq 0$, então

$$v_{(4i-2)d+\lceil (\ell+a)/2 \rceil} \in \{v_{(4i-2)d+\lceil \ell/2 \rceil-1}, v_{(4i-2)d+\lceil \ell/2 \rceil}; v_{(4i-2)d+\lceil \ell/2 \rceil+1}\}.$$

Assim, qualquer que seja o movimento do espião, o i -ésimo guarda pode ir de $v_{(4i-2)d+\lceil \ell/2 \rceil}$ para $v_{(4i-2)d+\lceil (\ell+a)/2 \rceil}$ ou se movendo para um de seus vizinhos, ou permanecendo estacionário.

- Finalmente, se $\ell * (\ell + a) < 0$, então $(\ell, a) = (-1, 2)$ ou $(\ell, a) = (1, -2)$. Neste caso, o i -ésimo guarda permanece em $v_{(4i-2)d}$.

Em todos estes casos as propriedades se mantêm válidas após o movimento dos guardas.

□

Agora que já examinamos o parâmetro $d_k^s(G)$ quando o grafo G é um caminho iremos exibir resultados semelhantes para o caso em que G é um ciclo.

O teorema a seguir é consequência direta dos dois lemas seguintes.

Teorema 5.4 *Para todo ciclo C com $n + 1$ vértices e quaisquer $k \geq 1$, $s \geq 2$, temos*

$$\left\lfloor \frac{(n-1)(s-1)}{k(2s+2)-4} \right\rfloor \leq d_{s,k}(C) \leq \left\lceil \frac{(n+1)(s-1)}{k(2s+2)-4} \right\rceil.$$

Lema 5.8 *Para todo ciclo C com $n + 1$ vértices e quaisquer $k \geq 1$, $s \geq 2$, temos*

$$d_{s,k}(C) \geq \left\lfloor \frac{(n-1)(s-1)}{k(2s+2)-4} \right\rfloor.$$

Prova: Novamente, apresentamos a prova para $s = 2$ para facilitar a leitura.

Seja $C = (v_0, v_1, \dots, v_n)$ e $d = \lfloor \frac{n-1}{6k-4} \rfloor$. A estratégia do espião é a seguinte. inicialmente, o espião ocupa o vértice v_0 e um guarda, denotado por g_0 , ocupa v_{-d} or v_{-d-1} or v_{-d-2} após o movimento dos guardas (os índices dos vértices devem ser lidos em módulo $n + 1$). Note que tais posições sempre podem ser atingidas: O espião fica a distância $d + 1$ do guarda g_0 e após o turno dos guardas, g_0 está a distância $d, d + 1$ ou $d + 2$ do espião. Então, a cada turno $i \geq 1$, o espião se move de v_{2i-2} para v_{2i} .

Provamos por indução em $1 \leq i < k$ que, após o turno $2id$, ou pelo menos $i + 1$ guardas estão ocupando os vértices em $\{v_{-d-2id-2}, \dots, v_{(4i-1)d-1}\}$, ou houve um turno $0 \leq j \leq i$ no qual, após o movimento dos guardas, a distância entre o espião e todos os guardas era pelo menos d .

Inicialmente, deve existir pelo menos um guarda, denotado por g_1 , ocupando um vértice em $\{v_{-d+1}, \dots, v_{d-1}\}$ caso contrário o espião estaria a distância pelo menos d de cada guarda. Note que g_0 e g_1 são guardas distintos.

Portanto, após o turno $2d$, os guardas g_0 e g_1 estarão ocupando vértices em $\{v_{-3d-2}, \dots, v_{3d-1}\}$ e o espião estará ocupando v_{4d} . Portanto, a hipótese de indução é válida para $i = 1$.

Seja $1 \leq i < k - 1$ e assuma por indução que, após o turno $2id$, existem pelo menos $i + 1$ guardas ocupando os vértices em $\{v_{-d-2id-2}, \dots, v_{(4i-1)d-1}\}$. Ademais, pela definição da estratégia do espião, este estará ocupando o vértice v_{4id} .

Então, após o turno $2id$, deve existir pelo menos um guarda, denotado por g_{i+1} , ocupando algum vértice em $\{v_{(4i-1)d+1}, \dots, v_{(4i+1)d-1}\}$ caso contrário todos os guardas estariam à distância pelo menos d do espião no turno i . Portanto, após o turno $2(i + 1)d$, o guarda g_{i+1} estará ocupando um vértice em $\{v_{(4i-3)d+1}, \dots, v_{(4i+3)d-1}\}$ e o espião estará ocupando o vértice $v_{4(i+1)d}$. De modo similar, todos os $i + 1$ guardas que estavam ocupando vértices em $\{v_{-d-2id-2}, \dots, v_{(4i-1)d-1}\}$ após o turno $2id$ estão limitados a ocupar os vértices em $\{v_{-d-2(i+1)d-2}, \dots, v_{(4i+1)d-1}\}$ após o turno $2(i + 1)d$. Assim, a hipótese

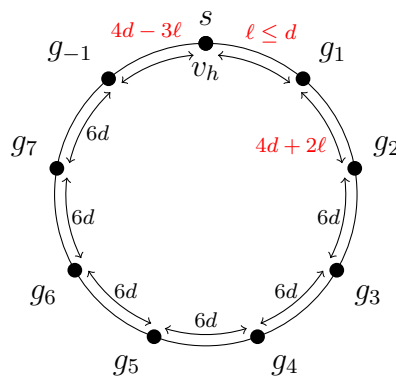
de indução é válida para $i + 1$: os guardas g_0, \dots, g_{i+1} estão ocupando vértices em $\{v_{-d-2(i+1)d-2}, \dots, v_{(4i+3)d-1}\}$.

Portanto, após o turno $2(k-1)d$, ou houve um turno anterior onde o espião esteve à distância pelo menos d de todos os guardas, ou todos os k guardas estão ocupando vértices em $\{v_{-d-2(k-1)d-2}, \dots, v_{(4k-5)d-1}\}$ enquanto o espião estará ocupando o vértice $v_{4(k-1)d}$. No último caso, se $v_{-d-2(k-1)d-2}$ está à distância pelo menos d de $v_{4(k-1)d}$ e $v_{4(j-1)d} \notin \{v_{-d-2(k-1)d-2}, \dots, v_{(4k-5)d-1}\}$ (em outras palavras, se $4(k-1)d + d \leq -d - 2(k-1)d - 2 \pmod{(n+1)}$), então o espião está à distância pelo menos d de todos os guardas neste turno. Observe que este é o caso, uma vez que $(6k-4)d < n$. \square

Lema 5.9 Para qualquer ciclo C com $n+1$ vértices e quaisquer $k \geq 1$ e $s \geq 2$,

$$d_{s,k}(C) \leq \left\lfloor \frac{(n+1)(s-1)}{k(2s+2)-4} \right\rfloor.$$

Figura 30 – Esquema de posições para o caso em que $k = 8$, $s = 2$



Fonte: Elaborado pelo autor.

Prova: Novamente, a prova abaixo considera que $s = 2$.

Observe que é suficiente provar o resultado para o caso em que $d = \frac{n+1}{6k-4} \in \mathbb{N}$. Let $C = (v_0, \dots, v_n)$. Novamente os índices dos vértices devem ser lidos em módulo $n+1$. Mostramos uma estratégia que garante que k podem manter um espião com velocidade s à distância no máximo d de pelo um guarda a cada turno (note que, na estratégia a seguir, o guarda g_1 está à distância $\ell \leq d$ do espião).

Inicialmente, o espião está em um vértice v_h para algum $0 \leq h \leq n$. Desejamos manter a propriedade que exista $0 \leq \ell \leq d$ tal que a configuração é a seguinte. Um guarda g_1 está em $v_{\ell+h}$, um guarda g_2 está em $v_{4d+3\ell+h}$, e um guarda g_{-1} está em $v_{-4d+3\ell+h}$. Então, para qualquer $3 \leq i \leq k-1$, um guarda g_i está em $v_{4d+3\ell+6d(i-2)+h} = v_{6di+3\ell-8d+h}$. Note que, para $3 \leq i \leq k-1$, o guarda g_i está à distância $6d$ do guarda g_{i-1} , e o guarda g_{k-1} está à distância $6d$ de g_{-1} . Mostramos agora como manter esta configuração independente do movimento do espião.

Obviamente, se o espião não se move, nenhum guarda muda de posição também.

Se o espião se move por uma aresta em sentido horário (respectivamente anti-horário), todos os guardas fazem o mesmo e a configuração também é mantida. Assim, podemos considerar apenas os casos em que o espião se movimenta por 2 arestas.

Basicamente, em todos os casos a seguir, o guarda g_1 executa seu movimento no mesmo sentido do espião o faz, enquanto todos os demais guardas fazem o seu movimento no sentido oposto.

- Caso o espião se mova para v_{h+2} (no sentido horário) e $\ell \geq 1$. Então, g_1 também se move no sentido horário e todos os demais guardas se movem no sentido anti-horário. Iremos mostrar que as propriedades continuam valendo para $0 \leq \ell' = \ell - 1 \leq d$ e $h' = h + 2 \pmod{n+1}$. De fato, g_1 se move de $v_{\ell+h}$ para $v_{\ell+h+1} = v_{\ell'+h'}$. O guarda g_2 se move de $v_{4d+3\ell+h}$ para $v_{4d+3\ell+h-1} = v_{4d+3\ell'+h'}$. O guarda g_{-1} se move de $v_{-4d+3\ell+h}$ para $v_{-4d+3\ell+h-1} = v_{-4d+3\ell'+h'}$. Finalmente, para todo $3 \leq i \leq k-1$, o guarda g_i se move de $v_{6di+3\ell-8d+h}$ para $v_{6di+3\ell-8d+h-1} = v_{6di+3\ell'-8d+h'}$. Assim, a propriedade continua válida após o movimento dos guardas.
- Caso o espião se mova para v_{h-2} (no sentido anti-horário) e $\ell \leq d-1$. Então, g_1 também se move no sentido anti-horário e todos os outros guardas se movem no sentido contrário. De maneira similar ao item anterior é possível checar que a propriedade é válida para $0 \leq \ell' = \ell + 1 \leq d$ e $h' = h - 2 \pmod{n+1}$.
- Caso $\ell = 0$. Assuma que o espião se move no sentido anti-horário de v_h para v_{h-2} (o caso em que ele se move para v_{h+2} é simétrico). Então, g_1 vai em sentido anti-horário para v_{h-1} , e todos os outros guardas se movimentam em sentido horário. De modo similar aos itens anteriores, podemos checar que a propriedade ainda é válida para $\ell' = 1$ e $h' = h - 2$.
- Caso $\ell = d$. Assuma que o espião vai, em sentido horário, de v_h para v_{h+2} (o caso em que ele se movimenta para v_{h-2} é simétrico, com o guarda g_{-1} assumindo o papel do guarda g_1). Então, g_1 vai, em sentido horário, para v_{h+d+1} , e todos os outros guardas se movimentam em sentido anti-horário. De modo similar aos itens anteriores, podemos checar que a propriedade é válida para $\ell' = d - 1$ e $h' = h + 2$.

□

5.5 Jogo do espião em grafos P4-tidy

Assim como fizemos para os demais problemas examinados neste texto, iremos utilizar as decomposições apresentadas na Seção 2.3 para obter algoritmos polinomiais para calcular $g_d^s(G)$ quando G é um grafo com poucos P_4 's.

Iniciamos este estudo determinando o valor de $g_d^s(G)$ quando G é obtido pela união de dois outros grafos.

Lema 5.10 *Se $G = G_1 \cup G_2$, então para todo $s > 1$ e $d \geq 0$ temos que $g_d^s(G) = \max\{g_d^s(G_1), g_d^s(G_2)\}$ na versão em que o espião se posiciona primeiro e $g_d^s(G) = g_d^s(G_1) +$*

$g_d^s(G_2)$ na versão em que os guardas se posicionam primeiro.

Prova: Na versão em que o espião se posiciona primeiro, se menos que $\max\{g_d^s(G_1), g_d^s(G_2)\}$ guardas forem usados, o espião pode se posicionar no grafo que necessita de mais guardas para ser protegido e utiliza sua estratégia vencedora para se comunicar com seu país. Caso $\max\{g_d^s(G_1), g_d^s(G_2)\}$ guardas estejam disponíveis, independente da escolha inicial do espião podemos posicionar um número suficiente de guardas de forma ótima no grafo escolhido e jogarmos utilizando a estratégia vencedora dos guardas para impedir que o espião se comunique.

Na versão em que os guardas se posicionam primeiro, caso sejam posicionados menos que $g_d^s(G_1)$ guardas em G_1 ou menos que $g_d^s(G_2)$ guardas em G_2 , o espião simplesmente se posiciona na componente com guardas insuficientes e utiliza sua estratégia vencedora para se comunicar. Assim, para vencer, $g_d^s(G_1)$ guardas devem se posicionar inicialmente em G_1 e $g_d^s(G_2)$ guardas devem se posicionar em G_2 , de forma que, independente de qual subgrafo o espião escolher para se posicionar, os guardas possuam uma estratégia vencedora para impedir o espião de se comunicar indefinidamente. \square

Nos lemas seguintes não há diferenças entre as estratégias dos guardas ou dos espiões nas versões em que os guardas se posicionam primeiro e em que o espião se posiciona primeiro. Isto é, em ambas as versões tanto guardas quanto espião jogam de maneira igual após a rodada 0.

Além disso, vale a pena salientar que, quando a velocidade do espião é igual a 1, não faz sentido analisarmos a versão em que o espião se posiciona antes dos guardas, uma vez que um guarda será sempre suficiente, bastando que este se posicione inicialmente no mesmo vértice do espião e permaneça sempre se movimentando da mesma forma que o espião durante o decorrer do jogo. Ademais não iremos analisar o caso em que $d = 0$ e $s = 1$ na versão em que o espião se posiciona depois, pois, como já mencionado anteriormente, este caso é equivalente a o jogo de polícia e ladrão que já foi analisado no capítulo anterior.

Lema 5.11 *Dados $G = G_1 \vee G_2$ e inteiros $d \geq 0$ e $s > 1$, então*

$$g_d^s(G) = \begin{cases} 1, & \text{se } d \geq 1 \text{ ou se } G \text{ for uma clique;} \\ 2, & \text{se } d = 0, s \geq 2 \text{ e } G \text{ não é uma clique.} \end{cases}$$

Prova: Se $d \geq 1$, então um guarda é suficiente. Basta que ele sempre se mantenha no subgrafo oposto ao do espião. Isto é, se o espião se movimentar para G_1 o guarda se movimenta(ou permanece) para um vértice de G_2 e vice versa.

Observe que no caso em que $d = 0$ e $s \geq 2$ a única possibilidade de 1 guarda poder impedir o espião de se comunicar é se G for uma clique. Senão, admita que G não é uma clique. Como G é conexo então deve existir um vértice v que possui distância exatamente 2 a pelo menos um outro vértice u . Se houver apenas um guarda, no momento em que o espião ocupar o vértice v o guarda também deve ocupá-lo, no entanto no turno seguinte o espião pode rapidamente, pois tem velocidade 2, se movimentar para u ,

enquanto o guarda pode se movimentar no máximo para um vizinho de u . Assim o espião pode se comunicar livremente.

Resta-nos mostrar que se $d = 0$, $s \geq 2$ e G não é um grafo completo, então dois guardas são o suficiente para proteger G . Neste caso, a cada rodada um dos guardas estará ocupando o mesmo vértice do espião no subgrafo G_i enquanto o outro estará em um vértice qualquer do subgrafo oposto. Se o espião escolher se manter em G_i o guarda que estava no subgrafo oposto se movimenta de maneira a ocupar o mesmo vértice do espião, enquanto o outro se movimenta para um vértice do subgrafo oposto. Se o espião escolher se movimentar para o subgrafo oposto o guarda que estava no mesmo vértice que ele se movimenta novamente para o mesmo vértice do espião enquanto o outro guarda se movimenta para algum vértice de G_i . \square

Resta-nos analisar os casos em que G é uma quase-aranha magra ou quase aranha gorda. Diferente do que fizemos no capítulo anterior não iremos separar os resultados de quase-aranhas e aranhas.

Lema 5.12 *Dados uma quase-aranha magra G com partição (R, C, S) e inteiros $d \geq 0$ e $s > 1$, então*

$$g_d^s(G) = \begin{cases} 1, & \text{se } d \geq 1; \\ k + 1, & \text{se } d = 0 \text{ e } R \neq \emptyset; \\ k + 1, & \text{se } d = 0, s \geq 3 \text{ e } \exists i \text{ tal que } s'_i \in S \text{ e não é adjacente à } s_i; \\ k, & \text{se } d = 0, R = \emptyset \text{ e } \forall i \text{ } s'_i \notin S \text{ ou é adjacente } s_i. \end{cases}$$

Onde k é a quantidade de vértices de C sem contar um possível vértice duplicado.

Prova: Iremos começar pelo caso em que $d \geq 1$ utilizando apenas um guarda. Observe que em uma quase-aranha magra se um guarda está posicionado em um vértice $c_i \in C$ os únicos vértices a distância maior que 1 do guarda são $\{s_j | j \neq i\}$ e um possível vértice duplicado s'_j também com $j \neq i$ ou um possível vértice duplicado c'_i (caso c_i e c'_i formem um conjunto independente). Se o espião se movimentar para um vértice s_j ou s'_j também com $j \neq i$ basta que o guarda se movimente para c_j . Se c_i e c'_i formarem um conjunto independente e o espião se movimentar para c'_i basta que o guarda se movimente para um outro vértice da clique.

Se $d = 0$, $s \geq 2$ e $R \neq \emptyset$, então uma estratégia vencedora para o espião jogando contra k guardas seria simplesmente ir para algum vértice de R , o que forçaria um guarda a se movimentar para o mesmo vértice. Como ficariam apenas $k - 1$ guardas em C , pelo menos um vértice s_i a distância 2 de qualquer guarda. No próximo turno, bastaria que o espião se movimentasse para s_i e dessa maneira poderia se comunicar com seu país de origem.

Se $k + 1$ guardas estiverem disponíveis a partir da clique, não importa para qual vértice o espião vá, um guarda pode segui-lo, enquanto os demais k policiais ocupam

cada um dos vértices de C , mantendo assim todos os vértices a distância no máximo 1 de pelo menos um guarda. Observe que, neste caso, o possível vértice c'_i não é ocupado por um guarda exceto no caso em que o espião está nele.

Se $s_i, s'_i \in S$ formando um conjunto independente, $d = 0$ e $s \geq 3$, então uma possível estratégia vencedora para o espião ao jogar contra k guardas seria ir para o vértice s_i , forçando um guarda a segui-lo. Se nenhum guarda se movimentasse para c_i bastaria que o espião, no próximo turno, se movesse para s'_i para se comunicar. Se algum guarda ocupasse c_i então, como só k guardas estão disponíveis, algum vértice s_j com $j \neq i$ ficaria a distância no mínimo 2 de qualquer guarda, como o espião possui velocidade 3 ele poderia simplesmente se mover para s_j e se comunicar no turno seguinte. A estratégia para os $k + 1$ guardas vencerem é análoga ao caso anterior.

Quando $d = 0$, $R = \emptyset$ e $s'_i \notin S$ ou é adjacente a s_i , então apenas k guardas são necessários uma vez que um vértice $s_i \in S$ com os vértices $\{c_j \in C \mid i \neq j\}$ formam um conjunto dominante de G . A estratégia dos guardas é análoga aos casos acima, exceto que quando o espião se movimenta para para um vértice s_j ou s'_j ele é seguido pelo guarda em c_j e c_j permanece desocupado até que o espião mude sua posição. \square

Lema 5.13 *Dados uma quase-aranha gorda G com partição (R, C, S) e inteiros $d \geq 0$ e $s > 1$, então*

$$g_d^s(G) = \begin{cases} 1, & \text{se } d \geq 1; \\ 3, & \text{se } d = 0, \text{ e } R \neq \emptyset; \\ 3, & \text{se } d = 0, \text{ e } \exists i \text{ tal que } s'_i \in S \text{ e não é adjacente à } s_i; \\ 3, & \text{se } d = 0, \text{ e } \exists i \text{ tal que } c'_i \in C \text{ e não é adjacente à } c_i; \\ 2, & \text{se } d = 0, \text{ e } R = \emptyset \text{ e } \forall i \ s'_i \notin S \text{ ou é adjacente à } s_i \text{ e } \forall j \ c'_j \notin C \text{ ou é adjacente a } c_j. \end{cases}$$

Prova: Iremos começar pelo caso mais simples em que $d \geq 1$. Ao posicionarmos um guarda em um vértice c_i de C ele está a distância 1 de todos os vértices do grafo exceto s_i e possivelmente de s'_i (se existir) ou c'_i (se existir e formar um conjunto independente com c_i). Caso o espião se movimente para s_i , c'_i ou s'_i , então basta que o guarda se movimente para algum vértice c_j com $j \neq i$.

Se $d = 0$ e $R \neq \emptyset$, então uma estratégia vencedora para o espião jogando contra 2 guardas seria simplesmente ir para algum vértice de R , o que forçaria um guarda a se movimentar para o mesmo vértice. Como restaria apenas 1 outro guarda em $C \cup S$, então ao menos um vértice s_i estaria a distância pelo menos 2 de todos os guardas. No próximo turno, bastaria que o espião se movimentasse para s_i e dessa maneira poderia se comunicar com seu país de origem.

Se 3 guardas estiverem disponíveis a partir da clique, não importa para qual vértice o espião vá, um guarda pode segui-lo, enquanto os demais 2 policiais ocupam cada um um vértice distinto de C , mantendo assim todos os vértices a distância no máximo 1 de pelo menos um guarda.

Se $d = 0$ e existe um vértice $s'_i \in S$ não adjacente à s_i , então uma possível estratégia vencedora para o espião ao jogar contra 2 guardas seria ir para o vértice s_i , forçando um guarda a segui-lo. Se o guarda restante não se mover para c_i , então haverá pelo menos um vértice s_j à distância pelo menos 2 de todos os guardas e no próximo turno bastaria que o espião se movesse para s_j para se comunicar com seu país. Caso o guarda restante se mova para c_i então ambos os guardas estariam posicionados em vértices não adjacentes à s'_i , bastaria então que o espião se movimentasse para s'_i no turno seguinte para se comunicar. A estratégia vencedora para 3 guardas é análoga ao caso anterior.

Se $d = 0$ e existe um vértice $c'_i \in S$ não adjacente à c_i , então uma possível estratégia vencedora para o espião ao jogar contra 2 guardas seria ir para o vértice s_i , forçando um guarda a segui-lo. Se o guarda restante não se mover para c_i ou c'_i , então haverá pelo menos um vértice s_j à distância pelo menos 2 de todos os guardas e no próximo turno bastaria que o espião se movesse para s_j para se comunicar com seu país. Caso o guarda restante se mova para c_i (c'_i) então ambos os guardas estariam posicionados em vértices não adjacentes à c'_i (c_i), bastaria então que o espião se movimentasse para c'_i (c_i) no turno seguinte para se comunicar. A estratégia vencedora para 3 guardas é análoga ao caso anterior.

Quando $d = 0$, $R = \emptyset$, $s'_i \notin S$ ou é adjacente à s_i e $c'_i \notin C$ ou é adjacente à c_i , então apenas 2 guardas são necessários uma vez que um vértice $s_i \in S$ com o vértice c_i formam um conjunto dominante de G . A estratégia dos guardas é análoga aos casos acima, exceto que quando o espião se movimenta para para um vértice s_j ou s'_j ele é seguido pelo guarda em c_j e o guarda restante se movimenta para c_i . \square

Teorema 5.5 *Dados um grafo P_4 -tidy G e inteiros $d \geq 0$ e $s \geq 1$ podemos determinar $g_d^s(G)$ em tempo $O(n + m)$ para ambas as versões do jogo do espião.*

Prova: Iremos utilizar a mesma técnica já aplicada nos capítulos anteriores. Primeiro, utilizando o Teorema 2.3, calculamos a árvore de decomposição de G , T_G , em tempo $O(m + n)$. Cada folha G' de T_G é um K_1 , C_5 , P_5 , $\overline{P_5}$ ou uma quase-aranha sem cabeça. No caso de G' ser um grafo quase-aranha sem cabeça, podemos calcular em tempo constante utilizando os Lemas 5.12 e 5.13. Os casos em que G' é um K_1 , C_5 , P_5 , $\overline{P_5}$ são trivialmente solucionados.

Cada vértice interno G' de T_G é um subgrafo grafo de G resultado de operações (\vee , \cup ou \uplus) sobre os seus filhos. Assim, se G' é o resultado de uma operação \cup podemos calcular $g_d^s(G')$ em tempo constante (assumindo que os filhos desde já foram resolvidos) utilizando o Lema 5.10. Se G' é o resultado de uma operação \vee podemos calcular $g_d^s(G')$ em tempo constante utilizando o Lema 5.11. Se G' é o resultado de uma operação \uplus (formando assim uma quase-aranha) podemos calcular $g_d^s(G')$ em tempo constante utilizando os Lemas 5.12 e 5.13.

Como T_G possui $O(n)$ vértices, todo o processo, até chegar na raiz G de T_G , leva tempo $O(n + m)$. \square

Utilizando estes resultados e as técnicas mostradas nos capítulo anterior seria possível obter um algoritmo polinomial para $(q, q - 4)$ -grafos com q fixo. No entanto, seria necessário, assim como existia para o jogo de polícia e ladrão, um algoritmo polinomial que decidisse se uma quantidade fixa de k guardas poderiam vencer o espião. Deixamos essa questão para trabalhos futuros.

6 CONCLUSÃO

Nesta tese de doutorado, apresentamos diversos resultados novos para os problemas de Coloração Localmente Identificável e do Jogo de Polícia e Ladrão. Além disso, introduzimos e também apresentamos diversos resultados para um novo jogo de perseguição em grafos que generaliza diversos outros jogos estabelecidos na literatura: o Jogo do Espião.

Provamos que determinar o número lid-cromático e o número slid-cromático é $O(n^{1-\varepsilon})$ -inaproximável em tempo polinomial, a menos que $P = NP$. Isso reforça a dificuldade do problema que já havia sido caracterizado como NP-completo por Esperet et al. (ESPERET *et al.*, 2012).

Apesar da dificuldade do problema de decidir se $\chi_{lid}(G) \leq k$ e $\widetilde{\chi}_{lid}(G) \leq k$, utilizando decomposições de grafos, pudemos mostrar algoritmos FPT para decidir estes parâmetros para $(q, q-4)$ -grafos (com q fixo) e para grafos com largura em árvore limitada (com a largura em árvore e k fixos). Além disso, melhoramos o limite superior para $\widetilde{\chi}_{lid}$ de grafos com grau máximo Δ para $2\Delta^2 - 3\Delta + 4$. O melhor limite superior conhecido anteriormente era de $2\Delta^2 - \Delta + 1$ (FOUCAUD *et al.*, 2012).

Em relação ao Jogo de Polícia e Ladrão, nos concentramos em desenvolver algoritmos para dois parâmetros do jogo, o *cop-number* e sua contraparte temporal, o *capture time*. Utilizando novamente decomposições de grafos, mostramos algoritmos polinomiais para determinar ambos os parâmetros para grafos P_4 -tidy e $(q, q-4)$ -grafos (com q fixo). Nossos resultados ainda implicam que a famosa conjectura de Meyniel é válida para grafos P_4 -tidy conexos e $(q, q-4)$ -grafos conexos com pelo menos q vértices.

Para o Jogo do Espião provamos que o problema é NP-difícil e $W[2]$ -difícil para toda velocidade $s \geq 2$ do espião e qualquer distância $d \geq 0$, a menos que $P = NP$. Ainda sobre a dificuldade do problema mostramos que uma versão do jogo em grafos direcionados é *PSPACE*-difícil em DAG's.

Devido a dificuldade do problema do Jogo do Espião nos voltamos então para abordá-lo em classes de grafos simples como caminhos e ciclos. Para estes grafos mostramos limites superiores e inferiores para $d_k^s(P_n)$ e $d_k^s(C_n)$. Apresentamos ainda um algoritmo linear para calcular $g_d^s(G)$ para quaisquer $s > 0$ e $d \geq 0$ quando o grafo G é um grafo P_4 -tidy.

Deixamos como problema em aberto determinar se é possível decidir, em tempo polinomial, se um número fixo de guardas pode proteger um grafo G contra um espião. Em caso afirmativo isso poderia, juntamente com os resultados apresentados neste texto, implicar um algoritmo polinomial para os $(q, q-4)$ -grafos de maneira semelhante ao que foi feito no Capítulo 4 para o jogo de polícia e ladrão.

Os resultados deste trabalho foram publicados em (COHEN *et al.*, 2018; MARTINS and SAMPAIO, 2018; COHEN *et al.*, 2016; MARTINS and SAMPAIO, 2015).

REFERÊNCIAS

- AIGNER, M.; FROMME, M. A game of cops and robbers. **Discrete Applied Mathematics**, v. 8, p. 1–11, 1984.
- ANDREAEE, T. Note on a pursuit game played on graphs. **Discrete Applied Mathematics**, v. 9, p. 111–115, 1984.
- ANDREAEE, T. On a pursuit game played on graphs for which a minor is excluded. **Journal of Combinatorial Theory**, v. 41, p. 37–47, 1986.
- BABEL, L.; KLOKS, T.; KRATOCHVÍL, J.; KRATSCH, D.; MULLER, H.; OLARIU, S. Efficient algorithms for graphs with few P_4 's. **Discrete Mathematics**, v. 235, p. 29–51, 2001.
- BABEL, L.; OLARIU, S. On the structure of graphs with few P_4 's. **Discrete Applied Mathematics**, v. 84, p. 1–13, 1998.
- BAIRD, W.; BONATO, A. Meyniel's conjecture on the cop number: a survey. **Journal of Combinatorics**, v. 3, n. 2, p. 225–238, 2012.
- BALISTER, P. N.; RIORDAN, O. M.; SCHELP, R. H. Vertex-distinguishing edge-colorings of graphs. **J. Graph Theory**, v. 42, p. 95–109, 2003.
- BAUMANN, S. A linear algorithm for the homogeneous decomposition of graphs. **Report No. M-9615, Zentrum für Mathematik, Technische Universität München**, 1996.
- BODLAENDER, H. L. Achromatic number is NP-complete for cographs and interval graphs. **Information Processing Letters**, v. 31, p. 135–138, 1989.
- BODLAENDER, H. L. A tourist guide through treewidth. **Acta Cybern**, v. 11, 1993.
- BODLAENDER, H. L. A linear time algorithm for finding tree-decompositions of small treewidth. **SIAM Journal on Computing**, v. 25, p. 1305–1317, 1996.
- BODLAENDER, H. L. Treewidth: Algorithmic techniques and results. **Lecture Notes in Computer Science**, v. 1295, p. 29–36, 1997.
- BONATO, A.; CHINIFOROOSHAN, E.; PRALAT, P. Cops and robbers from a distance. **Theoretical Computer Science**, v. 411, n. 43, 2010.
- BONATO, A.; GORDINOWICZ, P.; KINNERSLEY, B.; PRALAT, P. The capture time of hypercubes. **The Electronic Journal of Combinatorics**, v. 20, p. 24, 2013.

BONATO, A.; HAHN, G.; GOLOVACH, P.A.; KRATOCHVÍL, J. The capture time of a graph. **Discrete Mathematics**, v. 309, p. 5588–5595, 2009.

BONATO, A.; NOWAKOWSKI, J. The game of cops and robbers on graphs. **Student Mathematical Library**, v. 61, 2011.

BRAVO, R. S. F. **Partições em grafos com poucos P_4 's**. 2011. Tese de Doutorado – COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2011.

BURRIS, A. C.; SCHELP, R. H. Vertex-distinguishing proper edge-colorings. **J. Graph Theory**, v. 26, p. 73–83, 1997.

CERNÝ, J.; HORŇÁK, M.; SOTÁK, R. Observability of a graph. **Math. Slovaca**, v. 46, n. 1, p. 21–31, 1996.

CHAITIN, G. J.; AUSLANDER, M. A.; CHANDRA, A. K.; COCKE, J.; HOPKINS, M. E.; MARKSTEIN, P. W. Register allocation via coloring. **Computer Languages**, v. 6, n. 1, p. 47–57, 1981.

CHARTRAND, G.; ERWIN, D.; HENNING, M.A.; SLATER, P.J.; ZHANG, P. The locating-chromatic number of a graph. **Bull. Inst. Combin. Appl.**, v. 36, p. 80–101, 2002.

CLARKE, N. E.; MACGILLIVRAY, G. Characterizations of k -copwin graphs. **Discrete Mathematics**, v. 312, p. 1421–1425, 2012.

CLARKE, N. E.; NOWAKOWSKI, R. J. Cops, Robber, and traps. **Utilitas Mathematica**, v. 60, p. 91–98, 2001.

COHEN, N.; HILAIRE, M.; MARTINS, N.; NISSE, N.; PÉRENNES, S. Spy-Game on graphs. **Proceedings of the The Eighth International Conference on Fun with Algorithms**, 2016.

COHEN, N.; INERNEY, F. M.; NISSE, N.; PÉRENNES, S. Study of a Combinatorial Game in Graphs Through Linear Programming. **28th International Symposium on Algorithms and Computation (ISAAC 2017)**, v. 92, p. 1–13, 2017.

COHEN, N.; Martins, N. A.; INERNEY, F. M.; NISSE, N.; PÉRENNES, S.; SAMPAIO, R. Spy-game on graphs: Complexity and simple topologies. **Theoretical Computer Science**, 2018.

CORNEIL, D. G.; LERCHS, H.; STEWART, L. K. Complement reducible graphs. **Discrete Applied Mathematics**, v. 3, n. 3, p. 163–174, 1981.

CORNEIL, D. G.; PERL, Y.; STEWART, L. K. Cographs: recognition, applications

and algorithms. **Congressus Numerantium**, v. 43, p. 249–258, 1984.

CORNEIL, D. G.; PERL, Y.; STEWART, L. K. A linear recognition algorithm for cographs. **SIAM Journal on Computing**, v. 14, n. 4, p. 926–934, 1985.

DINUR, I.; STEURER, D. Analytical approach to parallel repetition. **STOC '14: Proceedings of the forty-sixth annual ACM symposium on Theory of computing**, ACM, p. 624–633, 2013.

ESPERET, L.; GRAVIER, S.; MONTASSIER, M.; OCHEM, P.; PARREAU, A. Locally identifying colorings of graphs. **Electr. Journal Combin.**, v. 19, n. 2, 2012.

FEIGE, U. A threshold of $\log n$ for approximating set cover. **J. ACM**, v. 45, n. 4, p. 634–652, 1998.

FOMIN, F. V.; GOLOVACH, P.; KRATOCHVÍL, J.; NISSE, N. Pursuing fast robber in graph. **Journal Theoretical Computer Science**, v. 411, p. 1167–1181, 2010.

FOUCAUD, F.; HONKALA; LAIHONEN, T.; PARREAU, A.; PERARNAU, G. Locally identifying colourings of graphs with given maximum degree. **Disc. Math.**, v. 312, p. 1832–1837, 2012.

Frankl, P. Cops and robbers in graphs with large girth and Cayley graphs. **Discrete Applied Mathematics**, v. 17, p. 301–305, 1987.

GAREY, M. R.; JOHNSON, D. S.; STOCKMEYER, L. J. Some simplified np-complete graph problems. **Theor. Comput. Sci.**, v. 1, n. 3, p. 237–267, 1976.

GAVENCIÁK, T. Cop-win graphs with maximal capture-time. **Discrete Mathematics**, v. 210, p. 1557–1563, 2010.

GIAKOUMAKIS, V. P_4 -laden graphs: a new class of brittle graphs. **Information Processing Letters**, v. 60, p. 29–36, 1996.

GIAKOUMAKIS, V.; ROUSSEL, H.; THUILLIER, H. On P_4 -tidy graphs. **Discrete Mathematics and Theoretical Computer Science**, v. 1, p. 17–41, 1997.

GODDARD, W.; HEDETNIEMI, S.M.; HEDETNIEMI, S.T. Eternal security in graphs. **J. Combin.Math.Combin.Comput.**, v. 52, p. 169–180, 2005.

GOLDSTEIN, A. S.; REINGOLD, E. M. The complexity of pursuit on a graph. **Theoretical Computer Science**, v. 143, p. 93–112, 1995.

GOLDWASSER, J. L.; KLOSTERMEYER, W. Tight bounds for eternal dominating sets in graphs. **Discrete Mathematics**, v. 308, p. 2589–2593, 2008.

GONÇALVES, D.; PARREAU, A.; PINLOU, A. Locally identifying coloring in bounded expansion classes of graphs. **Disc. App. Math.**, v. 161, n. 18, p. 2946–2951, 2013.

GRIGGS, J.R.; YEH, R.K. Labeling graphs with a condition at distance 2. **SIAM J. Discrete Math.**, v. 5, p. 586–595, 1992.

HAHN, G.; MACGILLIVRAY, G. A note on k -cop, l -robber games on graphs. **Discrete Mathematics**, v. 306, p. 2492–2497, 2006.

HENNING, M. A.; SWART, O. R.; SWART, H. C. Bounds on distance domination parameters. **J. Combin. Inform. System Sci.**, v. 16, p. 11–18, 1991.

HOÀNG, C. **Perfect graphs**. 1985. Tese de Doutorado – School of Computer Science, McGill University, Montreal, 1985.

IRVING, R. W.; MANLOVE, D. F. The b -chromatic number of a graph. **Discrete Applied Mathematics**, v. 91, p. 127–141, 1999.

JAMISON, B.; OLARIU, S. A tree representation for P_4 -sparse graphs. **Discrete Applied Mathematics**, v. 35, p. 115–129, 1992.

JAMISON, B.; OLARIU, S. P -components and the homogeneous decomposition of graphs. **SIAM Journal of Discrete Mathematics**, v. 8, p. 448–463, 1995.

KARP, R. M. Reducibility among combinatorial problems. **Complexity of Computer Computations**, p. 85–104, 1972.

KARPOVSKY, M. G.; CHAKRABARTY, K.; LEVITIN, L. B. On a new class of codes for identifying vertices in graphs. **IEEE Transactions on Information Theory**, v. 44, p. 599–611, 1998.

KINNERSLEY, W. B. Cops and Robbers is EXPTIME-complete. **J. Comb. Theory Ser. B**, v. 111, n. C, p. 201–220, 2015.

KLOKS, T. Treewidth, Computations and Approximations. **Lecture Notes in Computer Science**, v. 842, 1994.

KLOSTERMEYER, W.F.; MACGILLIVRAY, G. Eternal dominating sets in graphs. **J. Combin. Math. Combin. Comput.**, v. 68, p. 97–111, 2009.

KLOSTERMEYER, W.F.; MYNHARDT, C.M. Graphs with equal eternal vertex cover and eternal domination numbers. **Discrete Mathematics**, v. 311, n. 14, p. 1371–1379, 2011.

LINHARES-SALES, C.; MAIA, A. K.; MARTINS, N.; SAMPAIO, R. M. Restricted

coloring problems on Graphs with few P_4 's. **Annals of Operations Research**, v. 217, p. 385–397, 2014.

LU, L.; PENG, X. On Meyniel's conjecture of the cop number. **Journal of Graph Theory**, v. 71, p. 192–205, 2012.

MAMINO, M. On the computational complexity of a game of cops and robbers. **Theoretical Computer Science**, v. 477, p. 48–56, 2013.

MARTINS, N.; SAMPAIO, R. Inapproximability of the lid-chromatic number. **Electronic Notes in Discrete Mathematics**, v. 50, p. 121–126, 2015.

MARTINS, N.; SAMPAIO, R. Locally identifying coloring of graphs with few P_4 s. **Theoretical Computer Science**, v. 707, p. 69–76, 2018.

MEHRABIAN, A. The capture time of grids. **Discrete Mathematics**, v. 311, p. 102–105, 2011.

NOWAKOWSKI, R.J.; WINKLER, P. Vertex-to-vertex pursuit in a graph. **Discrete Mathematics**, v. 43, p. 235–239, 1983.

PRALAT, P. When does a random graph have constant cop number? **Australasian Journal of Combinatorics**, v. 46, p. 285–296, 2010.

QUILLIOT, A. Jeux et pointes fixes sur les graphes. **Thèse de 3ème cycle, Université de Paris VI**, p. 131–145, 1978.

ROBERTSON, N.; SEYMOUR, P. D. Graph minors. II. Algorithmic aspects of tree-width. **J. Algorithms**, v. 7, p. 309–322, 1986.

SLATER, P. J. R-Domination in Graphs. **Journal of Association for Computer Machinery**, v. 23, n. 3, p. 446–450, 1976.

STOCKMEYER, L. J.; MEYER, A. R. Word problems requiring exponential time. **STOC '73 Proceedings of the fifth annual ACM symposium on Theory of computing**, p. 1–9, 1973.

ZUCKERMAN, D. Linear degree extractors and the inapproximability of max clique and chromatic number. **STOC '06 Proceedings of the thirty-eighth annual ACM symposium on Theory of computing**, p. 681–690, 2006.